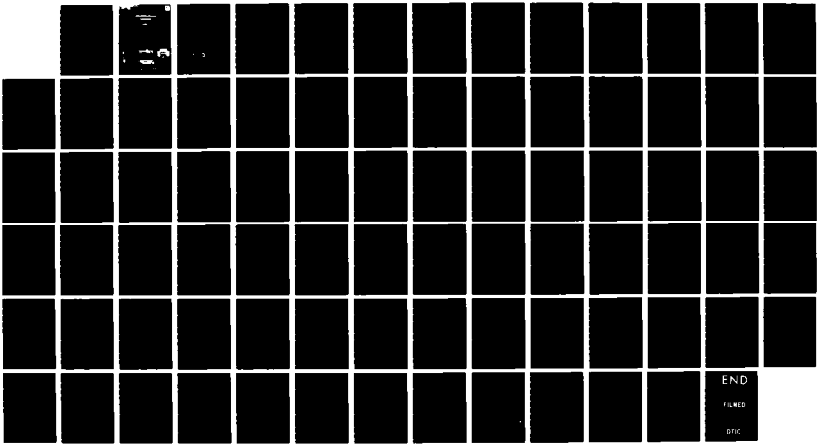
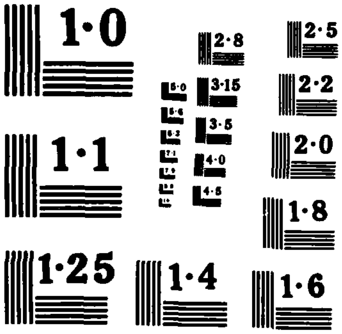
AD-A158 432 NETWORK REPAIR LEVEL ANALYSIS MODEL PROGRAMMER'S GUIDE 1/1
(U) AIR FORCE LOGISTICS COMMAND WRIGHT-PATTERSON AFB OH
NOV 84

UNCLASSIFIED F/G 9/4 NL





NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

FILE COPY

AD-A158 432

[illegible] 30 008

SUMMARY OF CHANGES

The Programmer's Guide was updated to include Appendix C, Additional Dimensioning Requirements. Appendix C presents additional information on dimensioning the program and is divided into three sections. Section 1 explains the additional dimensioning requirement resulting from implementation of Changes #5. Section 2 explains the dimensioning process using PARAMETER statements. Section 3 explains the dimensioning process without the use of PARAMETER statements.

Accession For
NTIS GRA&I ☑
DTIC TAB □
Unannounced □
Justification PER LETTER
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

DTIC
S ELECTE D
AUG 2 3 1985
B

DTIC
COPY
INSPECTED
3

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

CONTENTS

Page

LIST OF FIGURES

I. INTRODUCTION

1.1 Purpose

This programming guide has been written for the Network Repair Level Analysis (NRLA) Model computer program developed by AFALC/XRS at Wright-Patterson AFB, Ohio. Its purpose is to describe the program's structure, logic, input and output operations, and the organization of data used in the program so that modifications and/or corrections can be made. It is intended for use with the program and the NRLA User's Guide.

Appendix B, NRLA Program Array Dimensions, must be carefully read and understood by all users. Incorrect or incomplete redimensioning is the major user programming error.

1.2 Program Characteristics

The program is written in FORTRAN and was developed on the AFLC Honeywell 635 computer. Several FORTRAN features available (e.g., character variables, arithmetic expressions in DO statements, and quoted literals) were intentionally avoided in order to minimize the occurrence of incompatibilities with other FORTRAN compilers. The program has also been compiled, with minor changes, and executed on IBM hardware.

The program is composed of a main routine, a block data subroutine, plus 12 additional subroutines. Each of these is described in Chapter 2 with particular attention to data structures affected and CALLS to other subroutines. In addition, figures are provided to illustrate data structures and linkages between data elements.

1.3 Programming Conventions

As the computer program was being developed, particular emphasis was given to making the logic and code as straightforward and easy to comprehend as possible. Trade-off questions between programming ease versus logical simplicity were resolved by choosing simplicity. For example, all numeric data elements are stored in single dimensioned arrays rather than double or triple dimensioned arrays. Further, the array names were all chosen to be as descriptive as possible. Thus, the unit cost of an LRU is stored in array UCL, its mean time between failure is in array MTBF, the unit cost of an SRU is in UCS, and so on. These mnemonics promote the readability of the code and also preclude potential errors arising from using the wrong column of doubly subscripted arrays. Appendix A is a glossary of variable names.

A second programming convention relates to FORTRAN statement numbers. They were assigned, and sometimes reassigned, so that within each routine they would be in ascending order. Consequently, it is easy to find the destination for each GO TO and the terminal statement for each DO loop.

Another convention, used throughout the program, concerns DO loops. Many programs use a single letter variable for the DO loop parameters, as in

```
DO 10 I = 1,5 and
DO 20 I = J,K
```

These single letters may be reused for other loops or a change to double (or triple) letter variables may occur, e.g., II, JJ, JK, LMN, etc. The result is that it is frequently difficult to keep track of which loop most recently used a particular variable and therefore what the value of the variable is. This problem is avoided by using unique letter-number combinations for the loop parameters. The above loop examples would appear as

```
DO 10 I10 = 1,5 and
DO 20 I20 = J20,K20
```

Each variable will be a single letter followed by the statement number of the associated CONTINUE statement. This convention provides immediate traceability for letter-number variables and can be particularly beneficial for subscript values in long and/or nested loops

2. Program Logic

2.1 Introduction

This chapter provides detailed descriptions for each routine of the program. Its purpose is to present the logic and rationale in sufficient detail that the reader can follow and comprehend the program statements. It is not a line-by-line restatement of the program code; nor does it make gross statements of purpose describing 50 or more lines of code. It is intended to be a compromise between these extremes and still be comprehensive.

Despite this compromise, the reader will quickly realize that the amount of description is not always directly proportional to the amount of code being described. It has been assumed that the reader has substantial experience with FORTRAN programming; therefore, many details relating to FORTRAN "mechanics" and to elementary programming techniques are casually dismissed. For example, WRITE statements are mentioned but not detailed with regard to the number of statements, the constants/variables written, or the applicable FORMATs. Similarly, details are not given for data validation tests or tests preventing array overflow. Conversely, substantial explanation is given to some sections of code to explicitly clariŕy the "why" and/or "how" of the operations.

2.2 MAIN

The operation of the MAIN routine can be separated into five major functions:

a. Data input and verification,
b. LRU and SRU cost computations,
c. Display of item to SE relationships,
d. Network solution and sensitivity analysis, and
e. Output of results.

These functions are described separately, in sections 2.3 through 2.7, with particular attention given to a description of the data structures created.

2.3 Data Input and Verification

Figure 1 shows a sample NRLA input data file. The data values and relationships shown are used in subsequent figures to illustrate the data structures created by the program.

Data input and verification is accomplished by the statements from the beginning of the program through statement number 595.

The first three executable statements set certain parameters for two random access data files, 15 and 16, and obtain the current date, in the form YYMMDD. Modifications may be necessary to conform to the host computer's requirements. These are followed by initialization statements. Then, the input from file code 5 is read into the program. First, the output options and output units card is read. Next, the wholesale change and exclusion factors are read.

Next, the first input data record (stored on file code 10), Weapon System Program Factors, is read. The system name and run-identification data from it are used to write output report headers. Validity checks are done for the overseas deployment fraction and for sensitivity analysis paiameters with error messages written if appropriate. The remaining weapon system factors are then used to compute the monthly and life cycle flying hours.

```
MOD.THRT.EMT   47 020 10   1 120   10004      50 200 1
132 1642 136 3485 0.33 0.15
10420 10420 20.0 0.27 0.53 0.43 0.54 1.26 1.44 .389 .573 190.
11001 MULTIMETER        90. 1.80 1 2.5 200        0
12002 OSCILLOSCOPE    2375. 47.5 1 1.8 200        0
12003 SIGNAL.GEN.     4750. 95.0 1 1.2 200        0
12004 PULSE.GEN.      1350. 27.0 1 0.5 200        0
12005 POWER.SUPPLY     600. 12.0 1 2.5 200        0
12006 UNIV.BRIDGE     1375. 27.5 1 0.1 200        0
15001 MULTIMETER.*      90. 1.80 1 2.5 200        0
16002 OSCILLOSCOP*    2375. 47.5 1 1.8 200        0
16003 SIGNAL.GEN.*    4750. 95.0 1 1.2 200        0
16004 PULSE.GEN..*    1350. 27.0 1 0.5 200        0
16005 POWER.SUPP.*     600. 12.0 1 2.5 200        0
16006 UNIV.BRIDGE*    1375. 27.5 1 0.1 200        0
99
31 LRU05  SCR ASSY      1.  4552. 1.0 1.0 1.5 1.5 .33 .1  2  5000
31 LRU09  SWTCH SCR AY 1. 17670. 5.0 1.0 1.5 1.5 .33 .1  8  2500
31 LRU10  STOR CAP MAD 4.  2338. 0.5 1.0 1.5 1.5 .33 .1  4 15000
31 LRU14  PFN/SWT XFMR 1. 19810. 1.0 1.0 1.5 1.5 .33 .1  2  5000
99
41LRU05    1.67SRU53  CCA.CHRG  1  0    0. .01  6  2 .41 .41 .06 200.  3  0   .41
41LRU05    2.33                 1  0 683. .05  6  2 .80 .80 .06 200.  2  0   .80
41LRU09    1.17SRU54  DLAY.RCT  1  0    0. .10  6  2 .40 .40 .06 200.  1  0   .40
41LRU09    2.17SRU57  CS.TRANS  1  0    0. .01  6  2 .50 .50 .06 200.  1  0   .50
41LRU09    3.33SRU60  TTA.S.SC  1  0    0. .50  6  2 .55 .55 .06 200.  1  0   .55
41LRU09    4.33                 2  02651. .10  6  2 .53 .53 .06 200.  2  0   .53
41LRU10    11.0                 1  0 351. .01  6  2 .23 .23 .06 200.  5  0   .23
41LRU14    1.71SRU55  SWT.TRNS  1  0    0. .20  6  2 .80 .80 .06 200.  3  0   .80
41LRU14    2.25SRU56  PLS.FNET  1  0    0. .01  6  2 .55 .55 .06 200.  3  0   .55
41LRU14    3.04                 1  02972. .30  6  2 .40 .40 .06 200.  4  0   .40
99
51SRU53    698.  .5 209. .01  1  0  7 8 1.5 1.5 .33 .41 .41  6  2 .06 200.    .41
51SRU54    674. 2.0 202. .10  1  0  3 8 1.5 1.5 .33 .40 .40  6  2 .06 200.11 .40
51SRU57    544. 5.0 163. .01  1  0  4 2 1.5 1.5 .33 .50 .50  6  2 .06 200.1  .50
51SRU60   1500. 6.0   0. .50  1  0  510 1.5 1.5 .33 .58 .58  6  2 .06 200.    .58
51SRU55    956.  .5 287. .20  1  0  3 2 1.5 1.5 .33 .80 .80  6  2 .06 200.    .80
51SRU56   1628.  .2 488. .01  3  0  610 1.5 1.5 .33 .55 .55  6  2 .06 200.    .55
99
32 LRU05     10015001
32 LRU09     1001200220032005500160026003б005
32 LRU10     1001200650016006
32 LRU14     10015001
52 SRU53     10012002200320055001600260036005
52 SRU54     10012002200320055001600260036005
52 SRU57     10015001
52 SRU60     100120022003200420055001600260036004б005
52 SRU55     10015001
52 SRU56     1001200220032004200550016002600360046005
```

FIGURE 1. SAMPLE DATA FILE

Maintenance system factors are then read. The annual turnover rates are used to calculate life cycle turnover factors for maintenance personnel training.

Data values for the Supply Sy͗stem Factors are then read from the third input record.

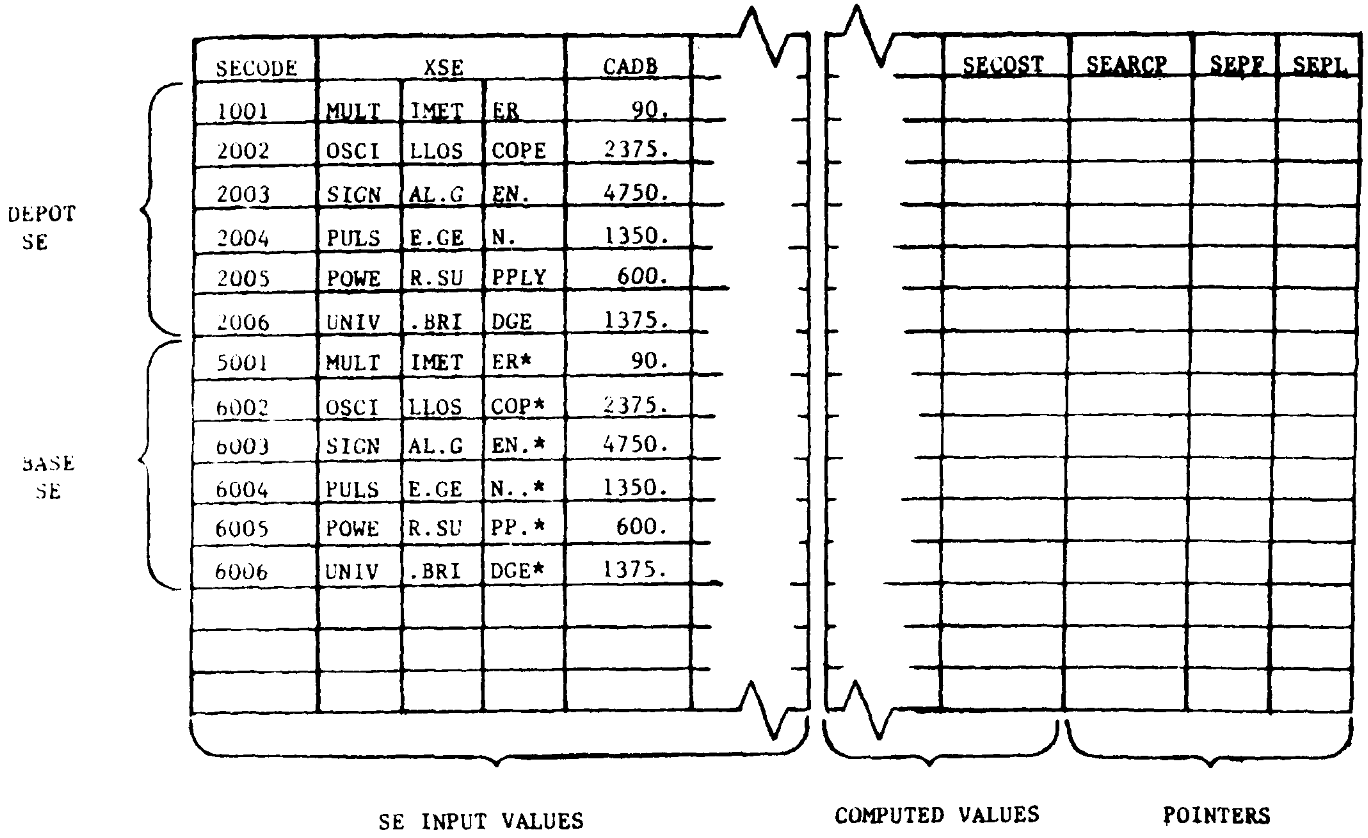
Values from the above data records are printed for user verification. This is followed by computations using the supply system factors and the printing of header information for support equipment data.

Support equipment input data is processed starting at statement number 200 and continuing up to statement 300. The variable NUMSER serves as a counter for the number of SE resources input and as a subscript for the SE data arrays. If the number of SE data cards exceeds the allocated array size (specified by the value for MAXSE) the excess cards are read, printed with an error message, and ignored. When array sizes will not be exceeded, the data is read into arrays of the common area SEIN and validation tests are performed for the first two input characters. The arrays SRCE, DEST, CAP, and FLOW serve as temporary storage areas for the digits of the SE code numbers. Each valid input record is printed to allow verification by the user. The SE input values plus certain computed values and pointers are stored in arrays as shown in Figure 2. A "row" of data will later be referred to as an SE data record and will exist for each depot and base SE. The functions of the computed values and pointers will be explained later.

The processing of LRU input data is performed from statement 300 to statement 350. The variable NLRU is used as a counter for the number of LRUs identified in the input file and as a subscript for LRU data arrays. Overflow of LRU arrays is prevented by comparing NLRU to the control value MAXLRU. The input value NSE, number of SE resources required for LRU repair, is subtracted from zero and the result saved in the array NSERL. This negative value is replaced by a positive value when the LRU to SE relationships are determined. LRU input data arrays and associated pointer arrays are shown in Figure 3. (NOTE: Entries are shown for TFAILP, FRSTFM, and LASTFM even though they are not established until the LRU failure mode data has been read.)

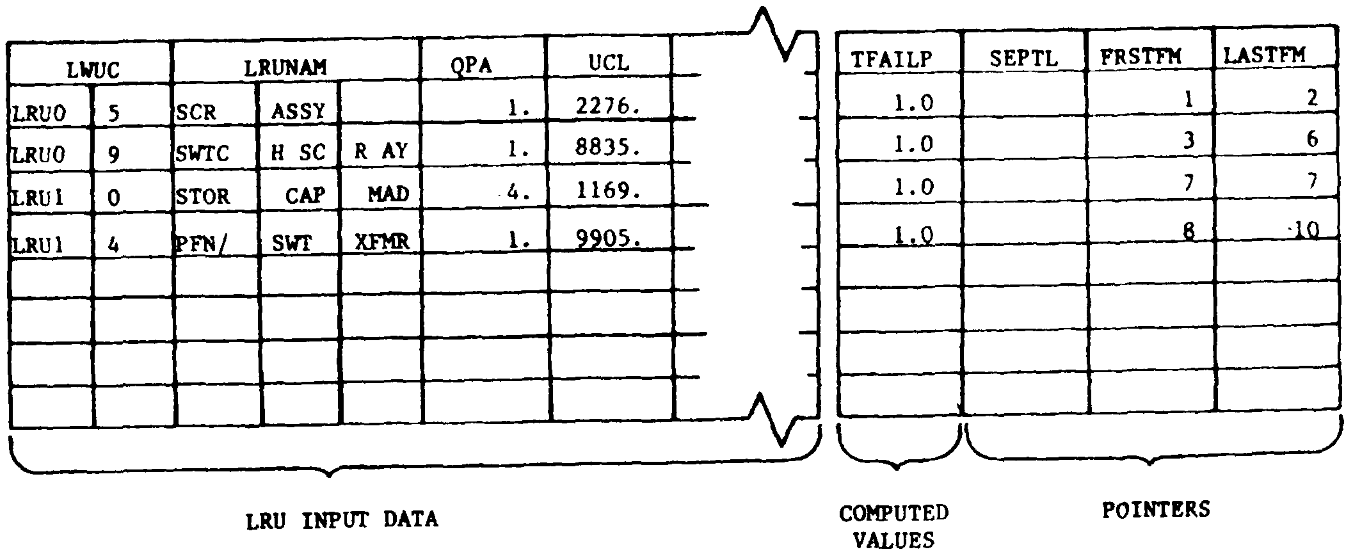
Input data cards for LRU failure mode data are processed from statement 400 up to statement 450. The variable LFMS is the counter for the number of failure mode data cards read and as a subscript for the LRU failure mode data arrays. Array overflow is prevented by comparing LFMS to the control value MAXFM. As each card is processed the corresponding LRU data record is located and the pointers FRSTFM and LASTFM updated. The input value NSE (number of SE required) is subtracted from zero and saved in the array NSERFM. Validity tests are done for the failure percentage input value and also for the values which serve to preclude specific repair level options (LFMOD, LFMOS, and LFMOB). If the input field for an SRU identifier is not blank the count of SRU data cards is incremented. This value, SFMS, is then used as a subscript to save the SRU identifier, the SRU name, and a pointer to the LRU failure mode record. In addition, the value SFMS is saved in the array SRUPTR as a pointer to the SRU data record. LRU failure mode arrays are shown in Figure 4.

SRU input data cards are read at statement 500 and processed prior to statement 550. Because there are no restrictions on the sequence of SRU cards in the input file, the SRU data values cannot be read directly into the arrays for SRU data records. The values are read into temporary locations and then moved to the proper SRU record based on matching the SRU identifier field. For each SRU the input values which preclude specific repair level options (SOD, SOS, and SOB) are validated and checked for consistency with options specified for the associated LRU failure mode. Figure 5 shows the arrays used for SRU data records.



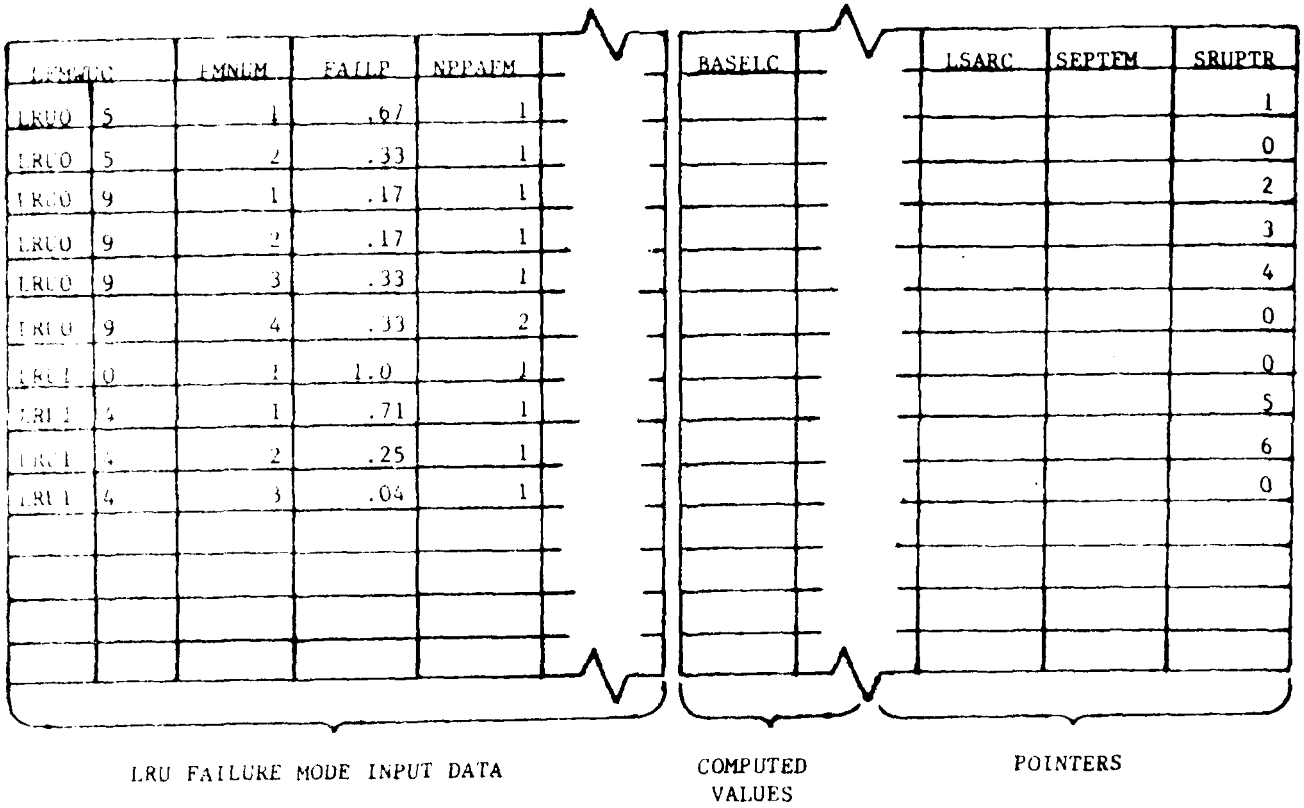
SECODE	XSE			CADB			SECOST	SEARCP	SEPF	SEPL
1001	MULT	IMET	ER	90.						
2002	OSCI	LLOS	COPE	2375.						
2003	SIGN	AL.G	EN.	4750.						
2004	PULS	E.GE	N.	1350.						
2005	POWE	R.SU	PPLY	600.						
2006	UNIV	.BRI	DGE	1375.						
5001	MULT	IMET	ER*	90.						
6002	OSCI	LLOS	COP*	2375.						
6003	SIGN	AL.G	EN.*	4750.						
6004	PULS	E.GE	N..*	1350.						
6005	POWE	R.SU	PP.*	600.						
6006	UNIV	.BRI	DGE*	1375.						

FIGURE 2. SUPPORT EQUIPMENT DATA STRUCTURE



LWUC		LRUNAM			QPA	UCL	TFAILP	SEPTL	FRSTFM	LASTFM
LRU0	5	SCR	ASSY		1.	2276.	1.0		1	2
LRU0	9	SWTC	H SC	R AY	1.	8835.	1.0		3	6
LRU1	0	STOR	CAP	MAD	4.	1169.	1.0		7	7
LRU1	4	PFN/	SWT	XFMR	1.	9905.	1.0		8	10

FIGURE 3. LRU DATA STRUCTURE



LRUWUC		FMNUM	FAILP	NPPAFM	BASELC	LSARC	SEPTEM	SRUPTR
LRU0	5	1	.67	1				1
LRU0	5	2	.33	1				0
LRU0	9	1	.17	1				2
LRU0	9	2	.17	1				3
LRU0	9	3	.33	1				4
LRU0	9	4	.33	2				0
LRU1	0	1	1.0	1				0
LRU1	4	1	.71	1				5
LRU1	4	2	.25	1				6
LRU1	4	3	.04	1				0

FIGURE 4. LRU FAILURE MODE DATA STRUCTURE

2.15 Subroutine RESET

RESET is called by MAIN, MTBFSA, and UCLSA. Its purpose is to exploit the known structure of the RLA network so that the computer processor time required to solve the network can be reduced. This is accomplished by examining certain source node to sink node paths and incrementing the allocated flow along the paths. This is done either for all LRUs and SRUs or for a specific LRU and its associated SRUs.

The choice between all LRUS and a specific LRU is controlled by the argument LPTR. If LPTR has the value zero it means that a feasible total flow must be assured (by setting the flow in every arc to zero) prior to incrementing the flow for every LRU failure mode and every SRU. A nonzero value means that a previously determined max-flow solution has been made non-optimal by increasing some arc capacity values associated with an LRU. This occurs during the sensitivity analysis process in MTBFSA and UCLSA. In this case RESET will only examine the arcs associated with the LRU being considered for sensitivity analysis.

For each LRU and SRU considered the subroutine utilizes the path from source node to depot node to base node to sink node (e.g., 1 to 4 to 6 to 10 in Figure 8). Direct pointers to the arcs are found in LDARC, LSARC, and LBARC for each LRU failure mode and in SDARC, SSARC, and SBARC for each SRU. For each arc in each path the difference between its capacity and flow values is computed. The minimum of these 3 values is added to the flow for each of the 3 arcs. The minimum value is used because it is the maximum amount by which the path flow can be increased and still have the flow in each arc less than or equal to its capacity.

2.16 Subroutine MAXFLO

This routine is called from MAIN, MTBFSA, and UCLSA. Its purpose is to determine the minimum cut for the network because the minimum cost set of repair level decisions is obtained directly from this cut. The solution technique used is a direct application of the two stage labelling procedure described on pages 17-22 of Flows in Networks by L.R. Ford, Jr. and D.R. Fulkerson, Princeton University Press, 1962.

The object of the first stage is to find a path from the source to the sink node along which the network flow can be increased. The second stage follows this path from the sink back to the source node modifying the allocated flow in each arc of the augmenting path. This two stage process is repeated until no flow augmenting path from source to sink can be found. When this occurs the minimum cut for the network has been determined.

Prior to initiating the labelling algorithm the loops DO 3005 and DO 3010 are executed. The first one is executed for each LRU failure mode and the second one for each SRU. In the first, the entries in LFMOD, LFMOS, and LFMOB are examined; a nonzero entry indicates a user specification to exclude the depot, scrap, and base options, respectively. The repair level decisions are precluded by setting the appropriate arc capacity to a very large number. The SRU loop operates similarly using the arrays SOD, SOS, and SOB as the source for the user's SRU specifications.

The first stage is then initiated by labelling the source node, setting its STATE to 2 (labelled and unscanned), and setting the STATE for every other node to 1 (unlabelled). At statement 3030 a loop is executed to find a labelled and unscanned node.

Arcs to node 6 are located by using the 5th and 6th entries of BACKSP. The 5th entry is related to the last triple with DEST equal to 5. Similarly, the 6th entry is related to the last triple with DEST equal to 6. Therefore, the appropriate pointers for node 6 are BACKSP (5) +1 through BACKSP (6). In this case, 8 and 9. These pointers are <u>not</u> subscripts for triples -- they are pointers to entries in BKPTR. The values in BKPTR are subscripts for the triples. The 8th and 9th entries in BKPTR are 8 and 13 indicating that the 8th and 13th triples have 6 as the DEST.

The last logical section of code in SETNET is the DO 2400 loop. Here the SRCE and DEST entries for each arc are examined to determine the function of the arc. If an arc is used for LRU costs a pointer to it is saved in LDARC, LSARC, or LBARC for LRU depot arc, LRU scrap arc, and LRU base arc, respectively. Similarly, pointers are saved in SDARC, SSARC, SBARC, SBDARC, and SBSARC for appropriate SRU arcs, and saved in SEARCP for SE costs.

2.14 Subroutine SORT

SORT is called from subroutine SETNET. As its name implies, its function is to sort values supplied by SETNET. Specifications to SORT are contained in its arguments list. The first three arguments to the routine (M1, M2, M3) are the arrays to be sorted, the fourth is an array for sequence pointers (LINK), the fifth specifies the number of values in the arrays to be sorted, the sixth specifies the dimensioned size of the arrays to be sorted, and the last argument is a 0-1 indicator directing SORT to only sequence the links or to also resequence the three arrays to be sorted. SORT is not a general purpose sorting routine. It only sorts into ascending order and sequences based only on the contents of the first argument array. Processing is essentially a two stage procedure.

The first stage begins by assigning the integers 1, 2, ..., N into array LINK. These integers are used as pointers to values in the array M1. The loop DO 30 then rearranges the LINK values so that they correspond to a sorted order for the values in M1. The first pass through the DO 30 loop results in moving the pointer to the largest value in M1 to the end of the LINK list. The second pass gets the pointer to the second largest M1 value to the second last entry of LINK list. Each pass gets at least one LINK pointer to its proper ordering position.

To clarify these operations, suppose that M1 had the 3 values 30, 20, and 10. LINK would first be initialized to 1, 2, 3. In the DO 30 loop the LINK values 1 and 2 would be used as subscripts to compare the values 30 and 20. Since 30 and 20 are not in ascending order the LINK values 1 and 2 are reversed to give the order 2, 1, 3. Next, the pointers 1 and 3 are used to compare the values 30 and 10; since they are out of order the LINK values are reversed to give the sequence 2, 3, 1. This puts 1, the pointer to the largest M1 value, last in the list. The next pass through the DO 30 loop would produce the LINK sequence 3, 2, 1. These values mean that the smallest M1 value is third, the next smallest is second, and the largest value is first.

Depending on the value of the 0-1 argument the subroutine will either RETURN or complete the second stage of processing. The second stage is the DO 50 loop in which the LINK entries plus the M1, M2, and M3 entries are rearranged. In the loop if a LINK value indicates that the M1, M2, and M3 values are out of sequence they are moved to the array TEMP. Then, the M1, M2, and M3 slots just vacated are filled with the proper values by using the LINK pointer; those vacated slots are then filled with their proper values; and so on until the values in TEMP can be moved to their proper slots. This continues until all LINKS and associated M1, M2, and M3 values are properly sequenced.

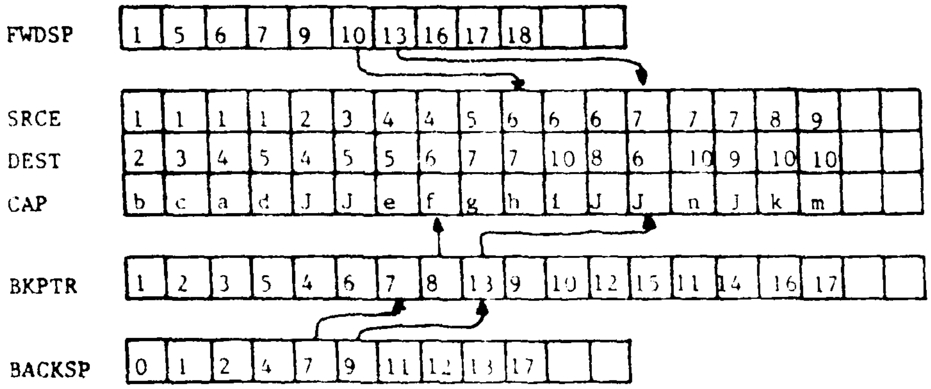
SRCE	1	1	1	1	2	3	4	4	5	6	6	6	7	7	7	8	9		
DEST	2	3	4	5	4	5	5	6	7	7	10	8	6	10	9	10	10		
CAP	b	c	a	d	J	J	e	f	g	h	i	J	J	n	J	k	m		

FIGURE 9.a. ARC ATTRIBUTE TRIPLES SORTED BY SRCE

FWDSP	1	5	6	7	9	10	13	16	17	18		

SRCE	1	1	1	1	2	3	4	4	5	6	6	6	7	7	7	8	9		
DEST	2	3	4	5	4	5	5	6	7	7	10	8	6	10	9	10	10		
CAP	b	c	a	d	J	J	e	f	g	h	i	J	J	n	J	k	m		

FIGURE 9.b. FWDSP VALUES AS POINTERS TO SRCE VALUES



FWDSP	1	5	6	7	9	10	13	16	17	18		

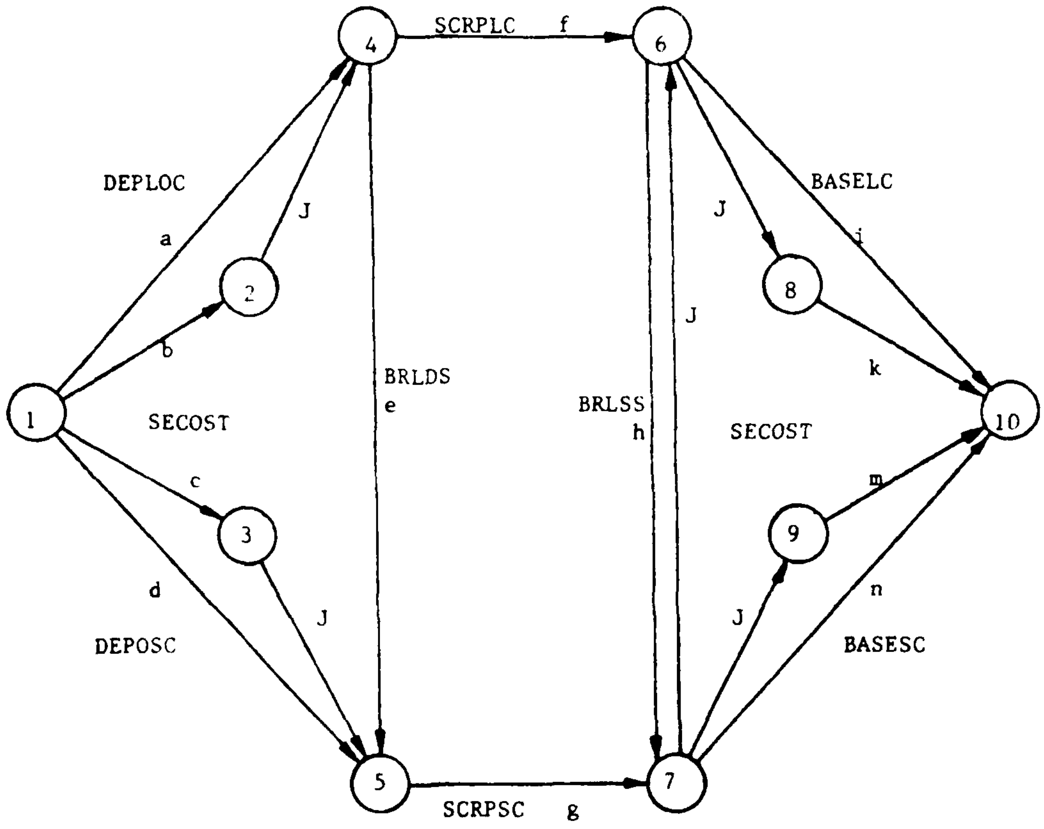
SRCE	1	1	1	1	2	3	4	4	5	6	6	6	7	7	7	8	9		
DEST	2	3	4	5	4	5	5	6	7	7	10	8	6	10	9	10	10		
CAP	b	c	a	d	J	J	e	f	g	h	i	J	J	n	J	k	m		

BKPTR	1	2	3	5	4	6	7	8	13	9	10	12	15	11	14	16	17		

BACKSP	0	1	2	4	7	9	11	12	13	17		

FIGURE 9.c. BACKSP AND BKPTR VALUES AS POINTERS TO DEST VALUES

FIGURE 9. NETWORK ARC ATTRIBUTES AND POINTER ARRAYS



SRCE	1	1	1	1	2	3	4	4	5	6	6	7	7	6	8	7	9		
DEST	2	3	4	5	4	5	5	6	7	7	10	6	10	8	10	9	10		
CAP	b	c	a	d	J	J	e	f	g	h	i	J	n	J	k	J	m		

FIGURE 8. SAMPLE NETWORK WITH ITS ATTRIBUTE TRIPLES

The DO 2120 loop assigns attributes for the two types of arcs originating at a base LRU node. One type goes to the corresponding SRU node and has a CAP value from BRLSS; the other goes to the sink node with a CAP value from BASELC. The loop is very similar to the DO 2080 loop.

Each base SRU node will ,e the source for two arcs, one arc to the appropriate LRU failure mode node and one to the sink node. These arcs are created in the DO 2130 loop.

The last set of arcs assigned are related to the base SE resources. The arrays SEXREF and NXTITM are used to assign arcs from LRU and SRU base nodes to the base SE nodes. This is analagous to the processing for depot SE in the DO 2060 loop. A second type of arc is created from each SE node to the sink node and assigned a CAP value from SECOST.

Figure 8 shows a sample network composed of arcs for one LRU failure mode, one SRU, two SE resources at depot, and two SE resources at base. Node numbers appear as they would be assigned in the program. With each arc is the array name from which the associated logistics cost is obtained and a lower case letter to represent the cost. The figure also shows the entries that would be made in arrays SRCE, DEST, and CAP for this sample network. The entries are in the order produced by the logic of subroutine SETNET.

To find the optimal solution for the network it is necessary to examine the CAP and FLOW values for each arc going out of a node and also those coming into the node. As shown in Figure 8 the attributes for arcs going from base LRU and SRU nodes (nodes 6 and 7) will not all be consecutive. Consequently, the attribute triples are sorted into ascending order by SRCE as shown in Figure 9.a. This order is beneficial because all arcs going out of node i can be examined by finding the first occurrence of i in SRCE and *continuing until the value in SRCE changes.* The benefit can be increased by saving a pointer to the first occurrence of each node rather than having to search for it. This is done in the DO 2190 loop with the pointers saved in FWDSP as shown in Figure 9.b. The loop DO 2195 is included to prevent errors which would arise if a node was not the source for any arc -- specifically, a depot SE not referenced by any LRU or SRU.

Because the solution algorithm needs to know which arcs go into each node, as well as those that go out, it would be beneficial to have the SRCE-DEST-CAP triples sorted by DEST. Unfortunately, the triples cannot be sorted by SRCE and also by DEST. However, a solution to this problem is obtained by including another array. A CALL is made to subroutine SORT with array parameters DEST and BKPTR. SORT will put the integers 1 through NARCS into BKPTR and then sort the integers, based on the contents of DEST, so that they can be used as pointers to DEST values. The loop DO 2200 uses the entries in BKPTR to build additional pointers in BACKSP. The results of these operations are shown in Figure 9.c. The loop DO 2222 is included to preclude problems which would arise if a node was not the destination for any arc -- a base SE not referenced by any LRU or SRU.

To clarify the utilization of FWDSP, BKPTR, and BACKSP consider node 6 of Figure 8. It is the source for 3 arcs and the destination for 2 arcs. The 6th entry of FWDSP is 10 indicating that the triples for arcs from node 6 start with the 10th triple. Based on the 7th entry of FWDSP, arcs from node 7, it is easy to determine that the appropriate triples for node 6 are referenced by subscripts 10 through 12 (i.e., 13 minus 1). Similar logic holds for every other node; the appropriate subscripts for node i are FWDSP (i) through FWDSP (i +1) -1.

2.13 Subroutine SETNET

SETNET is called by main after all costs and dependency relationships are known. The function of SETNET is to use this information to construct a matrix representation for the RLA network, determine and save pointers for use by the solution algorithm, and save pointers for use by the sensitivity analysis subroutines.

The matrix representation for the network is contained in arrays which specify the attributes for each network arc. These attributes are the source node for the arc, in array SRCE; the destination node for the arc, in DEST; the maximum flow (logistics cost) associated with an arc, in CAP; and the arc flow as later determined by the solution algorithm, in FLOW. The entries in the arrays are determined by using a predefined spatial structure for the nodes.

The fixed structure permits the determination of node numbers for key network nodes. This is done as the first step in the subroutine so that the values can be used for assigning SRCE and DEST node numbers. Nodes are assigned ordinal numbers, by node category, using the sequence: source node, depot SE nodes, depot LRU failure mode nodes, depot SRU nodes, base LRU failure mode nodes, base SRU nodes, base SE nodes, and sink node. The last depot SE (LDSE) node number will be 1 plus the number of depot SE (NDSE) resources. The last depot LRU (LDLRU) node number will be LDSE plus the number of LRU failure modes (LFMS). Similar logic is used to calculate node numbers for the last node in each of the node categories.

Assignment of arc attributes for depot SE arcs occurs in the loop DO 2010. Within this loop, and successive assignment loops, the variable NARCS is incremented as a counter for the number of network arcs and as a subscript for the arrays SRCE, DEST, and CAP. For each depot SE the value 1 is saved in SRCE, the SE node number is saved in DEST, and the life cycle SE cost is saved in CAP.

The next two loops, DO 2020 and DO 2030, assign the arc attributes for LRU failure mode and SRU repair costs at depot. These loops are essentially identical to the previous one.

Arcs from each depot SE node to the appropriate LRU and SRU nodes have their attributes assigned in the loop DO 2060. The SE to item arcs are generated stepping through the linked list of SE to item relationships saved in the SEXREF and NXTITM arrays. Positive values in SEXREF are pointers to LRU failure mode records. When the values are added to LDSE the result is the node number for the failure mode and is used for the arc's DEST entry. Negative values in SEXREF are pointers for SRU records. When these negatives are subtracted from the number of LRU failure modes (LFMS) the result can be added to LDSE to obtain the proper SRU node number to be saved in DEST.

The two types of arcs emanating from the depot LRU nodes have their attributes assigned in the loop DO 2080. For each failure mode with an associated SRU an arc is created from the LRU node to the SRU node and having a value from BRLDS assigned as its capacity. Then, arc attributes are saved to represent the arc from the LRU depot node to the LRU base node. This arc is assigned the LRU scrap option cost, from SCRPLC, as its CAP.

SRU scrap option costs are saved in CAP for the arc assignments made in the DO 2090 loop. These arcs have an SRU depot node as the source and an SRU base node as the destination.

2.11 Subroutine SRUCMP

This subroutine is very similar to FMCMP. It is called by MAIN, MTBFSA, and UCLSA to compute SRU related life cycle costs. The arguments LPTR, FMPTR, and SPTR serve as indices for locating the appropriate LRU, failure mode, and SRU data records.

Life cycle repair costs are computed as described in the NRLA User's Guide and saved in the SRU related arrays DEPOSC, SCRPSC, BASESC, BRLSS, and BRLDS. The last two correspond to the DEC2 and DEC1 costs described in the user's guide. Monthly maintenance man-hours required for SRU repair are computed and saved in SSEUHD and SSEUHB.

In relation to the computed SRU costs it is important to note the SRU scrap option costs are treated differently than the corresponding LRU failure mode costs. The LRU failure mode scrap costs are summed and assigned to a single arc in the network. The SRU scrap cost components are assigned to two different arcs so that each of the six joint LRU-SRU decisions will reflect the proper total cost. Specifically, the cost for packing and shipping SRUs plus the cost of base level inventory is assigned to be counted for a BASE-SCRAP decision combination but not counted for the DEPOT-SCRAP and SCRAP-SCRAP joint decisions. Further, to achieve the proper total for the SCRAP-SCRAP decision, the cost of SRU failures is subtracted from the corresponding LRU failure mode scrap cost.

2.12 Subroutine SECMP

SECMP is called by MAIN and by MTBFSA. It determines the quantity of each SE resource potentially required at depot and base level plus the life cycle cost for the resources. When called by MAIN the routine prints a table of SE requirement quantities, monthly utilization hours, and life cycle costs. The argument IND is set at zero by MTBFSA to suppress this printing.

At the beginning of the routine are write statements for output report identification and column headings. This is followed by a DO loop through statement 900. Instructions within the loop are executed for each SE resource.

The purpose of the code from statement 806 to statement 815 is to determine, for an SE, the expected monthly utilization from all LRUs and SRUs. The appropriate items are found by using an SEPF entry as a starting location pointer and stepping through the linked list of entries in SEXREF and NXTITM. Each entry of SEXREF is a pointer to an item record with positive values used for LRU pointers and negative values used as SRU pointers. The first digit of each SECODE indicates whether the resource is for depot (digits 1 to 4) or base (digits 5 to 8) use and therefore, whether the depot or the base SE use hours are totaled. The total utilization hours are then used to determine the SE quantity requirement.

This requirement is used to compute life cycle SE costs as described in the NRLA User's Guide. If SECMP was called from MAIN the computed hours, utilization rate, and costs are printed. If the call was from MTBFSA the new cost is assigned as the capacity value of the proper network arc.

from the previous decision to depot, scrap, and base, respectively. Of the 10 values, processing proceeds for those associated with decision change points. Actions include placing a "*" in array LINE corresponding to each negative value, changing negative values to positive, replacing each 4 value with the value preceeding it, and using the values as subscripts to place the values "DEPOT", "SCRAP", and "BASE" into the array LOCAT. The failure mode identifier is then written with the values in arrays LINE and LOCAT. If an SRU is associated with the failure mode the integer decision values for it are read from file 17 into array SADECS, processed as above to fill LINE and LOCAT, and an output line written to reflect its decision changes.

After all failure modes have been processed, two data records are read from file 17. The values are printed to specify the total number of LRU/SRU and SE decision changes per decision change point.

At statement 4190 the next record from file 17 is read into NDEC and DECVLO. A value for NDEC which is greater than zero indicates a continuation of decision change point information and causes a transfer back to statement 4040. If NDEC is not positive it will be -1 indicating the start of data for the next LRU. In this case the first 3 values of DECVLO are used for UCLSL, UCLSH, and UCLIN, respectively.

Statements from 4200 to 4300 process the SRU cost sensitivity results written to file 18 by subroutine UCLSA. Statements from 4300 to 4400 process the LRU MTBF sensitivity results written to file 19 by subroutine MTBFSA. These two sections contain logic and statements which are essentially identical to that just described for LRU cost sensitivity results.

2.8 Block Data Subroutine

This subroutine contains data statements to initialize values in the arrays CHARS and LOCAT.

2.9 Subroutine LRUCMP

LRUCMP is called by MAIN, MTBFSA, and UCLSA. Its function is to compute inventory stock levels and life cycle SCRAP option costs for an LRU. The argument LPTR is a pointer to an LRU data record.

The routine computes monthly and life cycle repair demand rates for the LRU. The monthly rate is used in equations which determine pipeline spares quantities and associated LRU inventory stock levels. The equations for these quantities and the life cycle SCRAP option costs are presented and described in Chapter 3 of the NRLA User's Guide.

2.10 Subroutine FMCMP

FMCMP is called by MAIN, MTBFSA, and UCLSA to compute repair level option costs for an LRU failure mode. The arguments LPTR and FMPTR are used as pointers to an LRU record and failure mode record, respectively.

The routine computes the life cycle logistics costs as described in the NRLA User's Guide and saves them in the arrays DEPOLC, SCRPLC, and BASELC. The expected number of maintenance man-hours required monthly for the failure mode is computed and saved in FSEUHD and FSEUHB for depot and base level repair, respectively.

The first action is a call to subroutine SETNET which constructs the network relationships. Subroutine RESET is then called to initialize the network flow. The minimum cost (maximum flow) solution for the network is determined by subroutine MAXFLO. Finally, subroutine OUTPUT is called to display the optimal SE and item decisions.

Prior to performing the sensitivity analysis computations, the program saves the contents of certain arrays which define the optimal solution. Specifically, STATE for node status plus CAP and FLOW for arc status. In addition, two random access files are established for temporary storage of data by the sensitivity analysis subroutines. A computed GO TO is then executed to transfer to statement 2025, 2035, or 2055 for LRU cost, SRU cost, or LRU MTBF sensitivity, respectively.

LRU cost sensitivity is done for each LRU with a call to subroutine UCLSA. In UCLSA the sensitivity results are written to a sequential binary data file, file code 17. In MAIN a REWIND is executed for the file and the first record read into memory.

The nested loops DO 2050, for each LRU, and DO 2040, for each failure mode, control SRU cost sensitivity. The analysis is done by UCLSA which writes the results to file 18. After statement 2050 the first record from file 18 is read.

For LRU MTBF sensitivity the loop DO 2060 is executed for each LRU. Subroutine MTBFSA is called to do the computations and write the results to file 19. After the loop, the first record of sensitivity results is read back into memory for subsequent display.

2.7 Output of Results

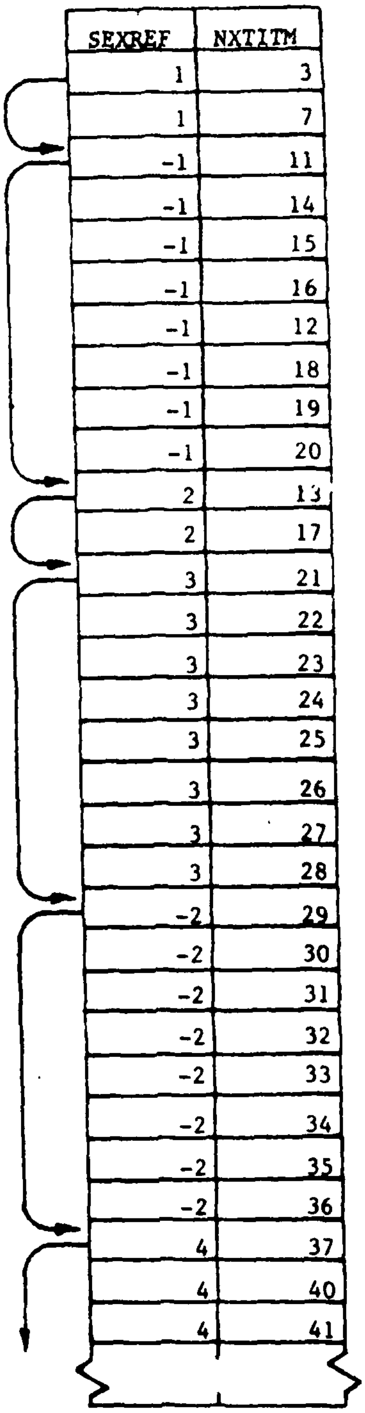
The display of repair level decisions details is done in the DO loop from statement 3000 through 5000. At the beginning of the loop are write statements which print the LRU input values and some computed failure rate values. Then, for each failure mode of the LRU, the loop DO 4000 is executed to print failure mode and SRU data. The program prints the input data factors, from failure mode and SRU data records, plus the computed logistics costs and the optimal repair level decision for the failure mode or SRU.

Statements between 4000 and 5000 control the writing of information resulting from the three types of sensitivity analysis. The statements prior to 4200 process the information on file 17, LRU unit cost sensitivity results.

In this section, the first action is to read a record from file 17 which indicates the number of decision change points, NDEC, and the low side LRU cost, DECVLO. If the number of decision change points is less than two a message is written indicating no decision changes. Otherwise, processing proceeds by writing a header line describing the input cost and sensitivity range plus two lines showing the cost values associated with the decision change points. The loop DO 4081 is executed to examine the cost values associated with each change point and from them determine where the set of optimal baseline decisions will print. A line is then printed which has "*****" positioned to be above the optimal baseline decisions.

The loop through statement 4160 is then executed for each failure mode of the LRU. A record is read from file 17 which contains 10 integer values each of which is -3, -2, -1, 1, 2, 3, or 4. The values provide coded information on decision changes. The values 1, 2, and 3 indicate the decisions depot, scrap, and base, respectively. The 4 indicates no change from the previous decision and the negative values -1, -2, and -3 indicate a change

SECODE	SEPF	SEPL
1001	1	85
2002	4	76
2003	5	77
2004	50	78
2005	6	79
2006	66	66
5001	2	86
6002	8	81
6003	9	82
6004	55	83
6005	10	84
6006	68	68



SEXREF	NXTITM
1	3
1	7
-1	11
-1	14
-1	15
-1	16
-1	12
-1	18
-1	19
-1	20
2	13
2	17
3	21
3	22
3	23
3	24
3	25
3	26
3	27
3	28
-2	29
-2	30
-2	31
-2	32
-2	33
-2	34
-2	35
-2	36
4	37
4	40
4	41

FIGURE 7. SECODE TO ITEM LINKAGES

The second section, from statement 620 through 630, is essentially identical to the first. The only difference is that it processes the SE requirements which are specified by failure mode.

At statement 635 the subroutine FMCMP is called to compute the life cycle logistic costs associated with each LRU failure mode. The total cost computed for each repair level option (base, depot, and scrap) is saved in an array. The individual cost equation values and the option totals are written to a temporary random file.

Finally, each failure mode is checked for an associated SRU. The processing for each SRU includes updating SEXREF and NXTITM to include the SRU's SE requirements, calling SRUCMP for cost computations, and saving the cost values in arrays and on a temporary random file.

Figure 7 shows the linkage from each SE through SEXREF and NXTITM to each item requiring the equipment. For SECODE 1001 the SEPF and SEPL entries specify that the linked list of items requiring the SE starts with the first SEXREF-NXTITM pair and ends with the 85th pair. The entries in NXTITM are the links to successive pairs. Thus, the first value in NXTITM, 3, points to the third pair; the third value, 11, points to the eleventh pair; the eleventh value points to the thirteenth pair; the 13th points to the 21st; the 21st to the 29th; and so on. The list terminates at pair 85 where the value of NXTITM is zero. The SEXREF values for this linked list are 1, -1, 2, 3, -2, 4, . . . The positive values are pointers to LRU failure modes which require SE resource 1001 for their repair. Similarly, the absolute value of each negative number is a pointer to an SRU requiring resource 1001.

2.5 Display of Item to SE Relationships

The printed output pages displaying the item to SE relationships are produced starting at statement number 750. Preliminary actions include initializing values and writing heading information. This is followed by a loop through statement 784 which is traversed for each LRU.

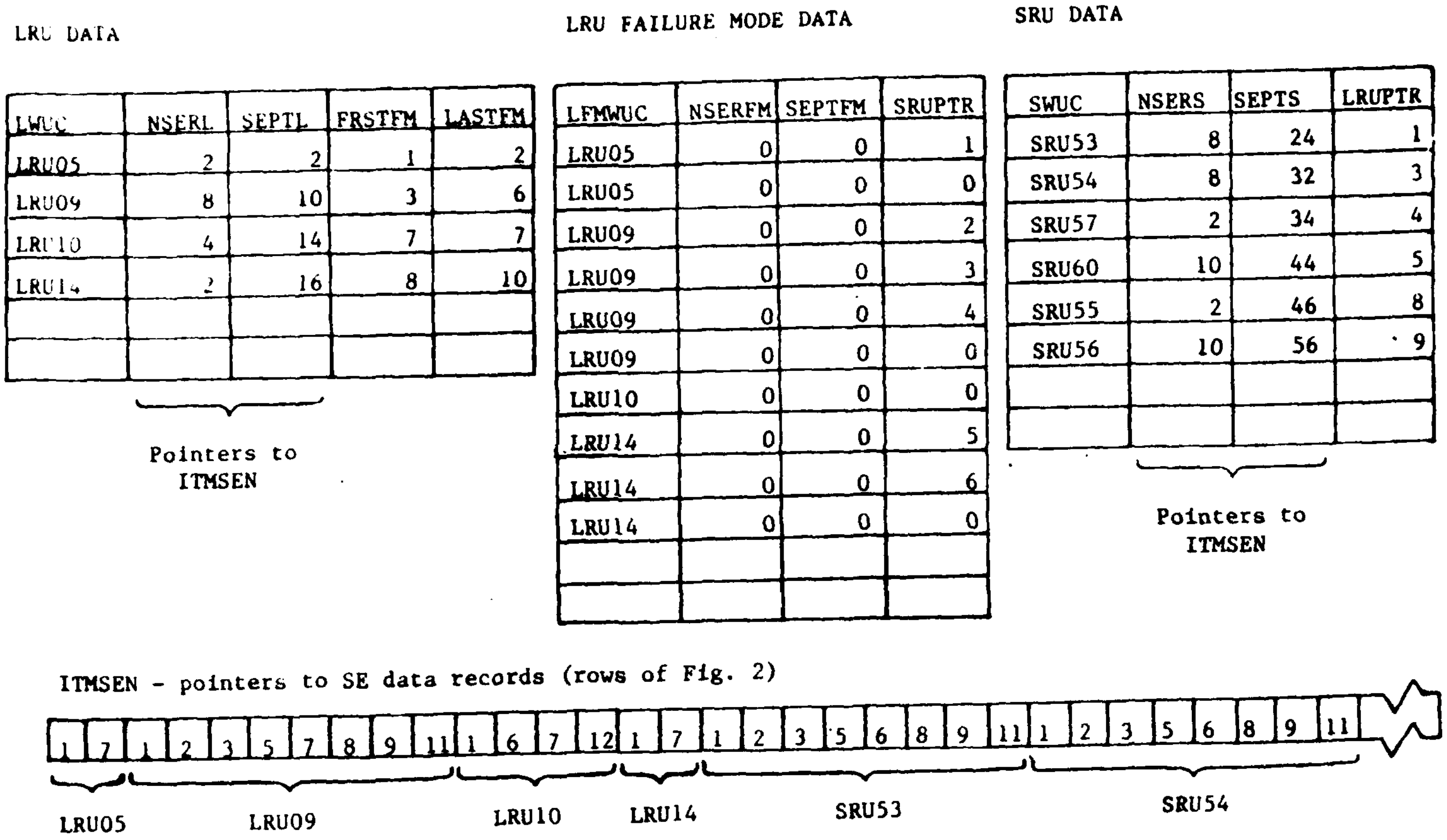
Within this loop is another, through statement 782, which is executed for each failure mode of each LRU. At the beginning of the inner loop are logical IF statements which control the writing of page headers and LRU related data. Then, by using the appropriate entries in the array ITMSEN, an X is placed in array LINE for each SE required for LRU repair. When the X's are then written out they are printed in a column under the associated SECODE number. The X's in LINE are then replaced by spaces and processing is done for an SRU.

SRU processing is similar to LRU processing in that a line of Xs is constructed and written out to reflect the SE requirements specified in ITMSEN.

After the item to SE relationships table has been written, the arrays SRCE, DEST, CAP, and FLOW, which were used for temporary storage of SE code numbers, are cleared to zero. Then, if any SE resource has no item references a message is printed to alert the user. Finally, subroutine SECMP is called to compute SE quantity requirements and life cycle costs.

2.6 Network Solution and Sensitivity Analysis

The network solution and sensitivity analysis function is programmed from statement 2000 to statement 3000.



LRU DATA

LWUC	NSERL	SEPTL	FRSTFM	LASTFM
LRU05	2	2	1	2
LRU09	8	10	3	6
LRU10	4	14	7	7
LRU14	2	16	8	10

Pointers to ITMSEN

LRU FAILURE MODE DATA

LFMWUC	NSERFM	SEPTFM	SRUPTR
LRU05	0	0	1
LRU05	0	0	0
LRU09	0	0	2
LRU09	0	0	3
LRU09	0	0	4
LRU09	0	0	0
LRU10	0	0	0
LRU14	0	0	5
LRU14	0	0	6
LRU14	0	0	0

SRU DATA

SWUC	NSERS	SEPTS	LRUPTR
SRU53	8	24	1
SRU54	8	32	3
SRU57	2	34	4
SRU60	10	44	5
SRU55	2	46	8
SRU56	10	56	9

Pointers to ITMSEN

ITMSEN - pointers to SE data records (rows of Fig. 2)

LRU05	LRU09	LRU10	LRU14	SRU53	SRU54
1 7	1 2 3 5 7 8 9 11	1 6 7 12	1 7	1 2 3 5 6 8 9 11	1 2 3 5 6 8 9 11

FIGURE 6. ITEM-TO-ITEM AND ITEM-TO-SE RELATIONSHIP LINKAGES

Next the data cards that specify the SE requirements for LRU and SRU repairs. The cards are read at statement 590 and the data processed by the statements from 350 to 400, 450 to 500, or 550 to 590 for LRU, LRU failure mode, and SRU requirements, respectively. The processing in each of the three sections is essentially identical. The input card is matched to the appropriate LRU, LRU failure mode, or SRU data record. Each SE identifier on the card is matched against the SECODE entries of the SE data records. For each match a pointer to the SE data record is saved in the next available element of the array ITMSEN and the appropriate LRU, LRUFM, or SRU record is updated to point to the new element of ITMSEN. In addition, the number of valid SE code matches is kept by the variable NSE for validation against the corresponding entry in NSERL, NSERFM, or NSERS.

Beginning at statement 595 the pareto change factors are read from file 5 and processed. If there are no pareto change factors, control is passed to statement 600.

Figure 6 illustrates the structural relationships between SE, LRU, LRUFM, and SRU data records developed during the data input and verification section of the program. The FRSTFM and LASTFM entries for LRU05 point to rows 1 and 2 of LRU failure mode data. Similarly, the failure mode records for LRU09 are in rows 3 through 6. The entries in the SRUPTR column contain pointers from the LRU failure mode records to the SRU records. The pointers from SRU records to LRU failure mode records are contained in the column LRUPTR. SE utilization input data is contained in the array ITMSEN. Each entry of ITMSEN is a pointer to an SE data record. For LRU05 the entry in NSERL indicates two SE references and the 2 in SEPTL indicates that the last of these SE pointers is the second entry of ITMSEN. The NSERL and SEPTL entries for LRU09 specify 8 SE pointers with the last one the tenth entry in ITMSEN, entries 3 to 10. The zeros in the NSERFM column indicate that the failure mode has no SE requirements other than those specified in the LRU record. SRU requirements for SE are specified in ITMSEN and referenced by NSERS and SEPTS.

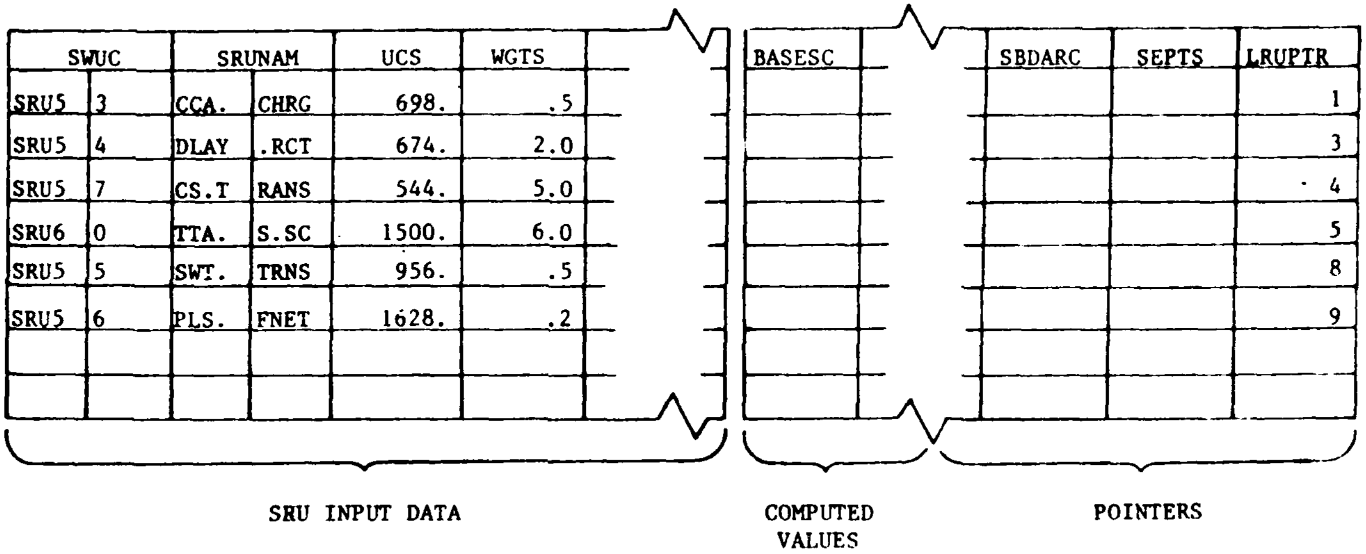
2.4 LRU and SRU Cost Computations

Cost computations and some related processing is accomplished by a DO loop from statement 600 through statement 690. The loop is executed for each LRU and within it is a loop through statement 680 which is executed for each LRU failure mode.

At the beginning of the outer loop the subroutine LRUCMP is called to compute inventory levels and life cycle costs for the LRU. The inner loop is then executed using the LRU related values in FRSTFM and LASTFM as pointers to the appropriate sequence of failure mode records.

At the beginning of the inner loop two sections of code are executed which accomplish an inversion of the LRU and LRU failure mode data in the ITMSEN array. At the beginning of the first section an arithmetic IF is done for the LRU's value in array NSERL. A negative value produces an error message indicating that an SE reference card was not supplied for the LRU; a zero value indicates no SE references for the LRU; and a positive value indicates the number of LRU SE references contained in ITMSEN.

As the entries in ITMSEN are successively processed the value NXREF is incremented to point to the next available entry of SEXREF (SE cross reference array) and NXTITM (next item in list). When the first item reference for an SE is encountered the SE arrays SEPF and SEPL will both be set to the value NXREF. As additional item references are processed, the value in SEPL is used to locate the LAST entry in the linked list, NXTITM at LAST is updated to point to the next reference (NXREF), and an item pointer is saved in SEXREF at NXREF.



SRU INPUT DATA
COMPUTED VALUES
POINTERS

SWUC		SRUNAM		UCS	WGTS		BASESC			SBDARC	SEPTS	LRUPTR
SRU5	3	CCA.	CHRG	698.	.5							1
SRU5	4	DLAY	.RCT	674.	2.0							3
SRU5	7	CS.T	RANS	544.	5.0							4
SRU6	0	TTA.	S.SC	1500.	6.0							5
SRU5	5	SWT.	TRNS	956.	.5							8
SRU5	6	PLS.	FNET	1628.	.2							9

FIGURE 5. SRU DATA STRUCTURE

This node is then processed (scanned) by examining every arc going out of this node and every arc going into the node. If the STATE of the node at the other end of an arc is 1 (unlabelled) then a label will be assigned to that node and it will become labelled-and-unscanned (STATE = 2) if:

a. the node is at the end of an outgoing arc and the arc flow is less than its capacity, or

b. the node is at the end of an incoming arc and the arc flow is greater than zero.

When all outgoing and incoming arcs have been processed the current node is assigned STATE = 3, labelled and scanned. At this point the STATE of the sink node is checked. If it is not labelled processing will be transferred to statement 3030 to find the next labelled-and-unscanned node.

A label on the sink node indicates that a flow augmentation path from the source to sink exists. This path is traversed in reverse order, from sink to source, by using the labelling information in the array NPATH. The labelling information is also used when computing a new flow for each arc in the path.

The two stage process, labelling and then augmenting, is repeated until no augmentation path can be found. This condition indicates that the maximum flow and the minimum cut have been found.

The flows in arcs out of the source node are then summed because it represents the total life cycle repair cost. The total cost for support equipment is also determined by examining the STATE associated with each depot and base support equipment node.

2.17 Subroutine OUTPUT

OUTPUT is called from MAIN to print the optimal solution results. This includes support equipment decisions plus LRU and SRU repair level decisions. Decisions are determined by using the STATE values assigned to nodes during the first stage of the labelling algorithm in subroutine MAXFLO. An arc of the network will be a member of the minimum cost cut for the network if the source of the arc is a labelled node and the destination node for the arc is an unlabelled node (see Flows in Networks, page 18). This property, combined with the fact that the source node for the network will always be labelled and that the sink node for the network will always be unlabelled, permits easy identification of SE and item decisions.

The first part of the routine prints the results for depot and base support equipment. A depot SE will be part of the minimum cost solution if its network node is unlabelled. Conversely, a base SE will be part of the minimum cost solution if its network node is labelled.

This is followed by the loop DO 3600 which is traversed for every LRU failure mode. For each failure mode the variables NODEL and NODER are used as pointers to the STATE for the left node (depot) and the right node (base), respectively. The two values in STATE are tested and a transfer to statement 3460, 3470, or 3480 is executed corresponding to depot repair, scrap, and base repair. Next, if the failure mode has no associated SRU a transfer is made to statement 3493, 3500, or 3510 for the proper WRITE statements. If there is an SRU the STATE of its left and right side nodes are teste:d and a transfer is made to 3530, 3550, or 3570 for the appropriate depot, scrap, and ba<e WRITE statements.

After the DO 3600 loop the totals for LRU, SRU, and SE costs are written and a RETURN executed.

2.18 Subroutine UCLSA

UCLSA is called from MAIN to determine the effects of LRU and SRU unit cost changes. The routine was originally written to determine if decision changes would occur as the LRU unit cost was varied across a range of values and to display the results of this investigation. The routine was modified, by adding logical IF statements and SRU cost computations, to determine and display the effects of SRU cost variations. Arguments to the routine include an LRU pointer, a failure mode pointer, a zero-one indicator for LRU versus SRU computations, plus factors specifying the low end and high end of the cost range to be investigated.

Processing in the routine is essentially the same for LRU and SRU analyses. For an LRU analysis the low end LRU cost is used to calculate new network arc costs and the new network is solved for optimum repair level decisions. Next, network arc costs and the optimum repair decisions are found using the high end LRU cost. Then, if the decisions at the low end, the middle, and the high end are identical appropriate messages are written to an output print file. If the sets of repair decisions are not identical then a binary search across the cost range is conducted to find all points at which repair decision changes occur. As each change point is found appropriate information is written to an output print file and also summarized on a temporary disk file for retrieval and printing by MAIN. The details for this processing are presented in subsequent paragraphs with the LRU versus SRU differences highlighted.

At the beginning of the routine are initialization statements for certain variables (e.g., NREC, NDEC, SAFILE, and LCRCMI) and WRITE statements to print the parameters for the analysis. The initialization and WRITE statements will be somewhat different for LRU versus SRU analyses. In addition, for an SRU analysis the subroutine arguments for the SRU cost range to be examined are used to compute a corresponding LRU cost range. Incorporating this range translation and a modification where costs are computed permits the SRU analysis to be accomplished with the LRU analysis logic.

Next, at statement 30 the low and high dollar values for the cost range examined are computed and written to file code 17, for LRU analyses, or file code 18, for SRU analyses. Then, the loop DO 39 is executed to initialize the values in SADECL, for each LRU failure mode, and in SADECS, for the corresponding SRUs. Each failure mode and each SRU has 10 values available to it in these arrays for recording decision changes during sensitivity analysis. The first of these values is initialized to 1, 2, or 3 to indicate that the optimal decision using the unmodified input factors was depot, scrap, or base, respectively. The remaining 9 values are initialized to 4 to indicate "no change in decision from the previous decision point". With this definition for 4 an entry into the arrays only needs to be made if a decision change actually occurs.

After these preliminary steps the sensitivity analysis investigation starts by examining the low end of the cost range. At statement 200 the variables SF and IRETRN are assigned in preparation for a transfer to statement 1000. At 1000 the LRU unit cost is modified by multiplying it by SF and subroutine LRUCMP is called to compute new values for LRU logistics costs. Then, the loop DO 1100 is executed to compute new logistics cost values for each LRU failure mode and for each associated SRU. Conditional logic is included in the loop so that the cost of each SRU is modified during an LRU analysis but the cost of only one SRU is modified during an SRU analysis. Also in the loop are statements to assign the new logistics costs to the appropriate arcs in the network structure. After the loop a transfer is made back to statement 110 where calls to RESET and MAXFLO occur. RESET is required because the newly computed logistics costs will be smaller than those they replace and, therefore, cause some arcs to have flow greater than capacity. The characteristics of the optimal solution found by MAXFLO (i.e., the state of each node plus the capacity and flow for each arc) are saved in the arrays LOSTAT, LOCAP, and LOFLOW by the loops DO 120 and DO 130. Also within these loops are statements to restore the characteristics of the baseline optimal solution (from the arrays OPSTAT, SAVCAP, and SAVFLO) and to determine if the low end decisions are different than the baseline decisions. If there are no decision differences a message is written out; otherwise, subroutine DECIDE is called to write messages stating the decision changes. Low end processing is completed by writing the total cost for the low end decisions and saving appropriate values in DECVLO and DECVHI.

Processing for the high end of the sensitivity cost range is done from statement 200 through 240. Parameters are set and a transfer to statement 1000 is made so that high end logistics costs can be computed and assigned to the associated network arcs. Upon return from the cost computation logic the subroutines RESET and MAXFLO are called to determine the high end optimal solution. High end computations are completed by saving the characteristics of the solution (in HISTAT, HICAP, and HIFLOW) and restoring the characteristics of the baseline solution.

The binary search for decision change points occurs from statement 300 to statement 400. The search begins with the loop DO 310 which tests for decision changes in two intervals -- between the low end and the middle, and between the middle and the high end. One or more change points will exist in an interval only if the decisions at the end points of the interval are different. Therefore, if the 3 sets of state values (in LOSTAT, STATE, and HISTAT) are identical there are no decision changes across the range of investigation. In this case an appropriate message is written and execution transfers to statement 470.

When decision change points do exist they could be in one interval or in both intervals. Since only one change point can be located at a time the latter situation is handled by building a push-down stack of search problems. Whenever it is determined that two intervals must be searched for change points the characteristics of the higher interval problem are written to the random access files 13 and 14. The problems on the files are later retrieved and solved using a last-in first-out criteria (i.e., a LIFO queue). When only the lower or upper interval contains a change point the width of the interval is compared to a predefined tolerance value, UCDLTA. If the width is less than UCDLTA a transfer to statement 400 occurs; otherwise, the interval width is bisected by using an LRU unit cost midway between the interval end point values. For the new LRU cost, logistics costs are computed and assigned utilizing the logic following statement 1000; RESET is called at statement 390; the optimal solution is found by MAXFLO; and a transfer back to statement 300 restarts the search for a decision change point.

As previously stated, processing commences at statement 400 when the interval containing decision changes is not wider than the tolerance width value. Processing includes writing a message which specifies the item cost values at the interval end points, saving the end point cost values in DECVLO and DECVHI, calling DECIDE to write the specific decision changes, and writing the total life cycle repair costs.

If another problem exists on files 13 and 14 it is retrieved and a transfer to statement 380 occurs to continue the search for decision change points. When no additional problems exist messages are written to indicate that no changes exist between the last change point located and the high end of the sensitivity cost range. Then, the loop DO 490 is executed to restore the arrays CAP and FLOW from SAVCAP and SAVFLO.

The last section of logic, from statement 510 through 600, writes summary information from the sensitivity investigation. The section is reached either when all change points have been found and detailed or when the eleventh change point has been found. In the latter case the summary information is written for the first ten points, to clear the arrays for additional data, and then processing transfers back to continue with the eleventh point. In either case the first summary data record written to SAFILE will specify the number of change points found (NDEC) and the item cost at the low end of the decision intervals. An IF test is done to determine if NDEC is 1. This value indicates that there is only one set of decisions across the entire cost range (no decision change points) and results in a RETURN back to MAIN. For NDEC greater than one, writes to SAFILE occur which specify, for each change point, the item cost at the high end of the decision intervals, the failure mode repair decisions, the SRU repair decisions, the number of LRU/SRU decision changes, and the number of SE decision changes. Then, if more than ten change points exist the arrays SADECL and SADECS are reinitialized and processing resumes at statement 416; otherwise, a RETURN to MAIN occurs.

2.19 Subroutine DECIDE

DECIDE is called from UCLSA and from MTBFSA. Its purpose is to specifically determine the decision changes identified by the sensitivity analysis routines and document the changes on an output print file. Both sensitivity routines contain two CALLs to DECIDE -- the first to document changes from the baseline decisions to the low end of the sensitivity range, and the second to document each change point located within the range of sensitivity investigations.

The two arguments IND1 and IND2 are each set to zero or one by the calling routine to control comparison processing within DECIDE. A value of one for IND1 means the decision changes between the low end and middle of an interval are to be documented, while a zero means changes between the middle and high end are to be documented. IND2 is one in the first call from UCLSA and MTBFSA to reverse the decision change direction, that is, a one means document changes from the middle to the low end rather than from low to middle (or middle to high) as is the case when IND2 is zero.

The routine is composed of three logical sections: one for depot SE changes, one for base SE changes, and one for LRU failure mode and SRU changes. The first two are short, straightforward, and essentially identical. The third is significantly longer but is also free of complexity. Each of the three sections contains dual sets of logic. One set uses the node state values in arrays LOSTAT and STATE, when IND1 is one; while the other set uses values in STATE and HISTAT, when IND1 is zero. Because the logic sets are identical in other respects the dual nature of each section will be ignored and the remainder of this description will present the logic for IND1 equal to 1.

The first section is the DO 30 loop which is traversed for each depot SE. Corresponding values in LOSTAT and STATE are compared. If the values are equal there is no decision change for the SE and, therefore, no further processing in the loop. Values which are different indicate differing SE requirements between the low end and middle decision solutions. In this case a message describing the change is written, a counter for the number of SE changes is incremented, and the loop begins for the next depot SE.

The DO 50 loop is used to document the decision changes for base SE. It is logically identical to the depot SE loop -- it tests for changes using LOSTAT and STATE, writes messages for the changes, and increments the count of changes.

The logic for LRU failure mode and SRU decision changes is in the DO 260 loop and is very similar to the logic for SE just described. The biggest difference is that the decision for an SE is determined from the state value for its single network node, but the decision for a failure mode or SRU is determined from the state values of its two nodes. For this reason, IF statements at the beginning of the loop test two LOSTAT values and then assign the value 1, 2, or 3 (for depot, scrap, or base) to the variable LOLVL. Similar tests on two STATE values result in a 1, 2, or 3 assignment to the variable MIDLVL. Now, if LOLVL and MIDLVL are equal there is no decision change for the failure mode; if they are different there is a change. Further processing in the loop is dependent on the existence of an SRU associated with the failure mode. If there is no SRU then either a message is written if there is a failure mode decision change, or processing continues with the next failure mode if there is no decision change. The existence of an SRU necessitates checking it for a decision change before proceeding. Logical IF tests using the SRU node values in LOSTAT and STATE result in the assignment of a 1, 2, or 3 to LOLVL and to MIDLVL. If an LRU and/or SRU decision change has occurred an appropriate message is written, a counter for the number of LRU and SRU changes is incremented, and the loop continues. After all failure modes and SRUs have been processed the count of item changes is saved in NLSCHG and a RETURN occurs.

2.20 Subroutine MTBFSA

MTBFSA is called from MAIN to determine the effects of changes in the MTBF for an LRU. The effects could be changes in failure mode repair level decisions, SRU repair level decisions, and/or changes to SE decisions. Arguments to the routine include a pointer to an LRU record and factors specifying the low end and high end of the MTBF range to be investigated. The logic flow and structure is essentially the same as in UCLSA. The differences exist primarily because UCLSA has dual sensitivity capability, LRU cost and SRU cost, and because there is an inverse relationship between MTBF and life cycle cost rather than the direct relationship between item cost and life cycle cost.

Just like UCLSA, the beginning of the routine has initialization statements for certain variables (e.g., NREC, NDEC, etc.), WRITE statements to document the subroutine's parameters, and initialization statements for the arrays SADECL and SADECS.

This is followed by statements to investigate sensitivity in the low end of the MTBF range. The logic parallels UCLSA in that new costs are computed and assigned by statements at the end of the routine; the new cost network is solved and its characteristics saved in LOSTAT, LOCAP, and LOFLOW; and solution differences are documented by DECIDE if there are any. The only significant difference from UCLSA is a call to SECMP. This is required because the lower MTBF value results in more item failures; thus, it could result in a higher SE requirement (e.g., 2 units instead of 1).

Next, the optimal solution for the high end of the MTBF range is found and its characteristics saved in HISTAT, HICAP, and HIFLOW. Once again, SECMP is called in case SE requirements have changed as a result of the new MTBF.

Statements from 300 to 500 contain the binary search logic for the decision change points. The procedure mirrors the corresponding section of UCLSA except that an additional test for changes in SE quantities is required. A change in quantities is signalled by a change in life cycle SE costs coupled with no decision changes.

Statement 500 is reached when a decision change point interval has been narrowed down to no larger than the predefined tolerance value DELTA. The decision changes are documented by subroutine DECIDE and the next interval to be searched is retrieved from files 13 and 14. When no additional problems exist to be solved a transfer to 600 occurs so that a final message can be written and the arrays CAP and FLOW restored.

Finally, the statements after 700 summarize the sensitivity results to file 19 just as UCLSA writes to files 17 and 18. Then, at 800 parameters are set so that LRU and SRU costs can be recalculated to reflect the input MTBF value. This is required because the cost computation routines also compute and save monthly maintenance man-hours which affect SE quantity requirements and are directly affected by LRU MTBF values.

Appendix A: Variables Dictionary

This appendix is primarily an alphabetical listing of FORTRAN variable names used in the NRLA program. The list contains dimensioning and descriptive information for the variables assigned to COMMON areas and for other significant variables.

Some program variables are not listed and described here. Some were omitted because their usage is restricted to only a few successive lines of code and the variable's function is clear from the context. For example, in the MAIN routine the variable C2DC2S is used for the sum of C2D and C2S and then used in a WRITE statement to print the sum. Other variables were omitted because the value and function are implied by the name. For example, the program uses ZERO for 0., MINUS1 for -1, PLUS1 for +1., and IMON, IDAY, IYR as integer values for the current date's month, day, and year.

For each variable in the list three columns of information are provided. The first column contains the variable name and, if applicable, dimensioning information below it within parentheses. The dimensioned size may be an integer constant, like (20) for CARD; an integer variable, like (MAXNOD) for BACKSP; or a combination of variable and constant, like (MAXLFM, 2) for LFMWUC. Where a variable appears as a dimension it should not be interpreted as "variable dimensioning". The proper interpretation is that the dimensioned size will vary among different applications but all arrays dimensioned with a particular variable need to be the same size. To clarify this point, the variable BASESC is shown with the dimensioning specification "(MAXSRU)". This indicates that BASESC needs to be dimensioned for the maximum number of SRUs in the program's input data file. Thus, BASESC may be dimensioned for 20 SRUs in one application but for 40 SRUs in another. In addition, regardless of the specific dimensioned size for BASESC it needs to be the same size as other SRU related arrays (BMMHS, BRLDS, etc).

The second column of information for each variable indicates the labelled COMMON area containing the variable (COSTF, MSDAT, etc.) or indicates the name of the subroutine(s) containing the variable if it is not in a COMMON area. Subroutine names are easily differentiated from COMMON names because they are indented by one space in the column. For some variables the second column value is "N/A" to indicate that the COMMON and subroutine classification are "not applicable" -- variable names used in many subroutines but not included in a COMMON area (I, J, K, etc.).

A definition for each variable is given in the third column of information. This column may also describe the function of the variable and will specify, for user input data values, the units associated with the variable (dollars, pounds, etc.) and the source data record for the value.

Variable	COMMON or SUBR.	Definition
A	COSTF	the expected annual cost for non-reparable assemblies and/or piece-parts required for repair of an LRU or SRU
A12	COSTF	the expected monthly cost for non-reparable assemblies and/or piece-parts required for repair of an LRU or SRU
AVHRS	SECMP	the expected number of hours per month that a particular type of currently installed support equipment will be available to support new workload of repair tasks
BAA	MSDAT	available work time per month for an intermediate level maintenance man (man-hours/month, Maintenance System Data Record)
BACKSP (MAXNOD)	NODDAT	an array of pointers, one for each network node, each of which points to an entry in BKPTR; the pointers are used to determine, by node, all arcs which terminate at each node
BASELC (MAXLFM)	FMCOMP	an array of values, one for each LRU failure mode, each of which is the sum of certain life cycle logistics costs associated with intermediate level repair of the LRU
BASESC (MAXSRU)	SCOMP	an array of values, one for each SRU, each of which is the sum of certain life cycle logistics costs associated with intermediate level repair of the SRU
BKPTR (MAXARC)	ARCDAT	an array of values, one for each network arc, each of which is a pointer to a network arc; the sequence of arcs specified by these pointers will be in ascending order by the arc's destination node number

Variable	COMMON or SUBR.	Definition
BLR	MSDAT	hourly labor rate for intermediate level maintenance men ($/hour, Maintenance System Data Record)
BMMHFM (MAXLFM)	LFMDAT	the number of maintenance man-hours required for repair of an LRU if the repair is done at intermediate level (man-hours/repair, LRU Failure Mode Data Record) Note: in Chapter 3 of the User's Guide this variable is BMMH
BMMHS (MAXSRU)	SRUDAT	the number of maintenance man-hours required for repair of an SRU if the repair is done at intermediate level (man-hours/repair, SRU Data Record)
BRCT (MAXLRU)	LRUDAT	the total elapsed time from removal of a failed LRU at a base, through intermediate level repair, until it is returned to serviceable base stock (months, LRU Data Record)
BRCTPL	LCOMP	the expected number of unserviceable LRU assets in the base repair pipeline
BRCTSL	LCOMP	the number of spare LRUs to be purchased to satisfy LRU demands expected to occur during the base repair cycle time
BRLDS (MAXSRU)	SCOMP	an array of values, one for each SRU, each of which is the logistics cost incurred if the next higher assembly LRU is base repaired and the SRU is depot repaired
BRLSS (MAXSRU)	SCOMP	an array of values, one for each SRU, each of which is the logistics cost incurred if the next higher assembly LRU is base repaired and the SRU is depot repaired or is scrapped at failure
BSYHRS (MAXSE)	SEIN	average monthly in-use time for a particular type of currently installed support equipment (hours/month/unit of SE, Support Equipment Data Record)

Variable	COMMON or SUBR.	Definition
C1	TEMPIN	integer value for the first character read from an input data record
C12	TEMPIN	integer value for the first two characters read from an input data record
C1B, C1D, & C1S	COSTF	the computed value for logistics cost category 1 for the base, depot, and scrap options, respectively; same variables used for LRUs and SRUs
C1SL	MAIN	temporary storage space for LRU scrap option cost so that the value is not destroyed when C1S is computed for an SRU
C2	TEMPIN	integer value for the second character read from an input data record
C2B, C2D, & C2S	COSTF	the computed value for logistics cost category 2 for the base, depot, and scrap options, respectively; same variables used for LRUs and SRUs
C2PREV	TEMPIN	temporary storage space for the previous value of C2
C2SL	MAIN	temporary storage space for LRU scrap option cost so that the value is not destroyed when C2S is computed for an SRU
C3B, C3D, & C3S	COSTF	the computed value for logistics cost category 3 for the base, depot, and scrap options, respectively; same variables used for LRUs and SRUs
C3SL	MAIN	temporary storage space for LRU scrap option cost so that the value is not destroyed when C3S is computed for an SRU

Variable	COMMON or SUBR.	Definition
C4B & C4D through C9B & C9D	COSTF	the computed values, for base and depot, for logistics cost categories 4 through 9, respectively; same variables used for LRU failure mode and SRU values
CADB (MAXSE)	SEIN	array of values, one for each SE, each of which is the unit cost of an SE for use at depot or base level ($, Support Equipment Data Record); NOTE: in Chapter 3 of the User's Guide this variable is UCSE
CAP (MAXARC)	ARCDAT	array of values, one for each network arc, each of which is the maximum flow capacity associated with the arc; the value is the logistics cost associated with the arc or, for the structural relationship arcs, it is a very large number (2**30)
CARD (20)	TEMPIN	array for temporary storage of a data card -- 20 words of 4 characters each
CHARS (15)	TEMPIN	array of character constants, defined in the BLOCK DATA subroutine, used in various WRITE statements
CODB (MAXSE)	SEIN	an array of values, one for each SE, each of which is the annual cost of operations and maintenance for a support equipment (dimensionless, Support Equipment Data Record) NOTE: in Chapter 3 of the User's Guide this variable is SEOPF.
DAA	MSDAT	available work time per month for a depot level maintenance man (man-hours/month, Maintenance System Data Record)
DECVHI (10)	SENSIT	an array of 10 values each of which is the high end boundary value for a range of values containing a decision change point; depending on the type of sensitivity each value will be an LRU cost, an SRU cost, or an MTBF

Variable	COMMON or SUBR.	Definition
DECVLO (10)	SENSIT	an array of 10 values each of which is a low end boundary value to match the values in DECVHI
DELTA	MTBFSA	maximum allowable difference between the end points of an interval containing a decision change point
DEPOLC (MAXLFM)	FMCOMP	an array of values, one for each LRU failure mode, each of which is the logistics cost associated with doing the LRU repair at the depot
DEPOSC (MAXSRU)	SCOMP	an array of values, one for each SRU, each of which is the logistics cost associated with doing SRU repair at the depot
DEST (MAXARC)	ARCDAT	an array of values, one for each network arc, each of which is the node number of the destination node
DEV	WSDAT	system support equipment development cost, a cost throughput (i.e., it does not affect the repair level decisions) ($1000, Weapon System Data Record)
DLR	MSDAT	hourly labor rate for depot level maintenance men ($/hour, Maintenance System Data Record)
DLTAFL (MAXNOD)	NODDAT	an array of values, one for each network node, each of which is used to record the potential for flow increase through the arc; a value is computed and assigned during each search for a flow augmenting path in subroutine MAXFLO
DMMHFM (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of wh ch is the number of maintenance man-hours required for repair of an LRU it the repair is done at the depot (man-hours/repair, *LRU* Failure Mode Data Record) NOTE: in Chapter 3 of the User's Guide this variable is DMMH
DMMHS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the number of maintenance man-hours required for repair of an SRU if the repair is done at the depot (man-hours/repair, SRU Data Record)

Variable	COMMON or SUBR.	Definition
DRCTC (MAXLRU)	LRUDAT	an array of values, one for each LRU, each of which is the elapsed time from removal of a failed LRU at a CONUS base until the item could become a serviceable spare *in depot stock*, it includes the time required for base to depot transportation and the depot shop flow time required for repair (months, LRU Data Record)
DRCTO (MAXLRU)	LRUDAT	an array of values, one for each LRU, each of which is the elapsed time from removal of a failed LRU at an overseas base until the item could become a serviceable spare in depot stock (months, LRU Data Record)
DRCTPL	LCOMP	the expected number of unserviceable LRU assets in the depot repair pipeline
DRCTSL	LCOMP	the number of spare LRUs to be purchased to satisfy LRU demands expected to occur during the depot repair cycle time
EOQ	COSTF	the dollar value of the base level inventory of non-reparable assemblies and/or piece-parts required for LRU or SRU repair
F1 through F12	MAIN	temporary storage locations for real values read from each SRU data card; values are transferred to the proper permanent arrays after the correct subscript is determined
FAILP (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is the LRU failure mode percentage, that is, the expected frequency for the type of failure as a fraction of all failures for the LRU (dimensionless, LRU Failure Mode Data Record)
FDB (MAXSE)	SEIN	an array of values, one for each SE, each of which is the cost of new facilities for support equipment ($, Support Equipment Data Record)

Variable	COMMON or SUBR.	Definition
FLOW (MAXARC + 2)	ARCDAT	an array of values, 2 greater than MAXARC, each of which is the amount of network flow through an arc, except the last 2 which are used for total flow and for total SE related flow
FMNUM (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is a user assigned 2 digit integer value used to distinguish the different failure modes for an LRU (dimensionless, LRU Failure Mode Data Record)
FMPTR	FMCMP SRUCMP	an integer arguement for the subroutines which is a pointer to the LRU failure mode for which computations are to be done
FMSEHR (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is the number of support equipmemt hours required for repair of an LRU (hours/repair, LRU Failure Mode Data Record)
FMTBCT	OUTPUT	for a failure mode, the mean time between corrective repair tasks
FPCT	OUTPUT	the FAILP value for a failure mode multiplied by 100 to print as a percent value
FRSTFM (MAXLRU)	LRUDAT	an array of values, one for each LRU, each of which points to the first set of failure mode data associated with an LRU
FSEUHB (MAXLFM)	FMCOMP	an array of values, one for each LRU failure mode, each of which is the expected number of support equipment utilization hours per month if repairs are done at the base; each value is the product of monthly failures times SE use hours per failure
FSEUHD (MAXLFM)	FMCOMP	identical to FSEUHB except for depot instead of base

Variable	COMMON or SUBR.	Definition
RIP (MAXLRU)	LRUDAT	an array of values, one for each LRU, each of which is a repair-in-place fraction; fraction of LRU failures which can be repaired at the organizational level, on-equipment maintenance (*dimensionless*, LRU Data Record)
RMC	SSDAT	recurring management cost to maintain an item in the wholesale inventory system ($/item/year, Supply System Data Record)
SA	SSDAT	annual cost to maintain an item in the base level supply system ($/item/year, Supply System Data Record)
SADECL (MAXLFM, 10)	SENSIT	an array of values, 10 for each LRU failure mode, each of which is a sensitivity analysis decision for the LRU at the decision change points
SADECS (MAXSRU, 10)	SENSIT	an array of values, 10 for each SRU, each of which is a sensitivity analysis decision for the SRU at the decisio change points
SAVCAP (MAXARC)	ARCDAT	an array of values, one for each network arc, used to save the arc capacity values from the baseline data
SAVFLO (MAXARC + 2)	ARCDAT	an array of values, 2 greater than MAXARC, used to save the network arc flow values from the optimum solution to the baseline problem, last 2 values are for total flow and for total SE related flow
SBARC (MAXSRU)	ARCPTR	an array of values, one for each SRU, each of which is a pointer to the SRU base repair cost arc
SBDARC (MAXSRU)	ARCPTR	an array of values, one for each SRU, each of which is a pointer to the network arc for costs incurred if the next higher assembly LRU is base repaired and the SRU is depot repaired

Variable	COMMON or SUBR.	Definition
PGMB	WSDAT	monthly end-item utilization hours at a base
PGMLCB	WSDAT	life cycle end-item utilization hours at a base
PGMLCS	WSDAT	life cycle end-item utilization hours for all bases
PGMS	WSDAT	monthly end-item utilization hours for all bases
PIUP	WSDAT	program inventory usage period; the life cycle of the system (years, Weapon System Data Record)
PSRC	SSDAT	packaging and shipping cost for CONUS shipments
PSRO	SSDAT	packaging and shipping cost for overseas shipments
PWRC	SSDAT	packed weight ratio for CONUS shipments; the ratio of the packaged item weight to the item weight (dimensionless, Supply System Data Record)
PWRO	SSDAT	packed weight ratio for overseas shipments (dimensionless, Supply System Data Record)
QPA (MAXLRU)	LRUDAT	an array of values, one for each LRU, each of which is the number of occurrences of the LRU in the end-item (No. LRU/end-item, LRU Data Record)
REQMT (MAXSE)	SECOMP	an array of values, one for each SE, each of which is the number of units of a support equipment that will have to be purchased to satisfy the repair requirements
RIIMC	SSDAT	the life cycle recurring plus initial item management cost

Variable	COMMON or SUBR.	Definition
OPHRS (MAXSE)	SEIN	an array of values, one for each SE, each of which is the expected time that a unit of SE will be available for item repairs (hours/month, SE Data Record)
OPSTAT (MAXNOD)	NODDAT	an array of values, one for each network node, each of which is the node's state from the optimal baseline solution determined by MAXFLO
OS	WSDAT	the fraction of the total number of end-items deployed to overseas locations (dimensionless, Weapon System Data Record)
OSTC	SSDAT	order and shipping time for CONUS locations; the elapsed time between the initiation of a request for a serviceable item from the depot and the receipt of the item at a CONUS base (months, Supply System Data Record)
OSTO	SSDAT	order and shipping time for overseas locations (months, Supply System Data Record)
OSTPL	LCOMP	the expected number of serviceable LRU assets in the depot to base order and shipping time pipeline
OSTSL	LCOMP	the number of spare LRUs to be purchased to satisfy LRU demands expected to occur during an order and shipping time cycle
PCC	SSDAT	packaging cost, including both labor and materials, for item shipments to CONUS bases ($/pound, Supply System Data)
PCO	SSDAT	packaging cost, including both labor and materials, for item shipments to overseas bases ($/pound, Supply System Data)

Variable	COMMON or SUBR.	Definition
NTBLFM (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is the minimum number of intermediate level maintenance personnel to be trained for the LRU repair task (number of people, LRU Failure Mode Data Record)
NTBS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the minimum number of intermediate level maintenance personnel to be trained for the SRU repair task (number of people, SRU Data Record)
NTDLFM (MAXLFM)	LFMDAT	same as NTBLFM except it's for depot personnel (number of people, LRU Failure Mode Data Record)
NTDS (MAXSRU)	SRUDAT	same as NTBS except it's for depot personnel (number of people, SRU Data Record)
NUMSER	SECOMP	number of SE resources in the input data file
NXREF	MAIN	pointer to the next available entry in the SEXREF and NXTITM arrays, equivalently, a counter for the number of entries in these arrays
NXTITM (MAXREF)	SEXDAT	an array of values each of which is either zero or a pointer to another entry in the array; the values are used with SEXREF
NXTSEN	MAIN	pointer to the next available entry in the ITMSEN array, also, the number of entries in ITMSEN
OPDECL (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is the optimal baseline decision for the failure mode
OPDECS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the optimal baseline decision for the SRU

Variable	COMMON or SUBR.	Definition
NSE	MAIN	input value indicating the number of SE resource identifiers associated with the current LRU, failure mode, or SRU (number of SE, LRU Data Record, LRU Failure Mode Data Record, or SRU Data Record)
NSECHG (10)	SENSIT	an array of 10 values, each of which is the total number of SE decision changes identified at a decision change point
NSECI (MAXSE)	SEIN	an array of values, one for each SE, each of which is the number of units of the SE currently installed at depot or base level (number of SE, SE Data Record) NOTE: in Chapter 3 of the User's Guide this variable is NSE
NSERFM (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which indicates the number of SE resources required for failure mode repair (number of SE, LRU Failure Mode Data Record)
NSERL (MAXLRU)	LRUDAT	an array of values, one for each LRU, each of which indicates the number of SE resources required for LRU repair (number of SE, LRU Data Record)
NSERS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which indicates the number of SE resources required for SRU repair (number of SE, SRU Data Record)
NSRU	MAIN	counter for the number of SRUs per LRU
NSRUS	SRUDAT	counter for the number of SRUs in the input data file

Variable	COMMON or SUBR.	Definition
NLRU	LRUDAT	counter for the number of LRU records in the user's data file
NLSCHG (10)	SENSIT	an array of 10 values, each of which is the total number of LRU and SRU decision changes identified at a decision change point
NNODES	NODDAT	total number of nodes *required for the network representation* of the user's data
NODEL, NODER	DECIDE, OUTPUT	integer pointers to the left side and right side nodes for an LRU failure mode or an SRU; used to determine and/or write out an item's repair level decision
NPATH (MAXNOD)	NODDAT	an array of values, one for each network node, used in MAXFLO to record the labelling path through the network *and then to traverse* backwards along the path
NPPAFM (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is the number of new (not in AF inventory system) piece-parts and/or assemblies required for LRU repair (No. items, LRU Failure Mode Data Record) NOTE: in Chapter 3 of the User's Guide this variable is NPPA
NPPAS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the number of new (not in AF inventory system) piece-parts and/or assemblies required for SRU repair (No. items, SRU Data Record)
NSAL	MAIN	during the output phase, the number of print lines required for sensitivity analysi; summary information; used to determine if page eject is necessary before writing the information

Variable	COMMON or SUBR.	Definition
NABFM (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is the number of standard (already stock numbered in AF inventory) items which will have to be entered into the base inventory system if LRU repair is done at the base (No. items, LRU Failure Mode Data Record) NOTE: in Chapter 3 of the User's Guide this variable is NAB
NABS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the number of standard (already stock numbered in AF inventory) items which will have to be entered into the base inventory system if SRU repair is done at the base (No. items, SRU Data Record)
NAME (2)	TEMPIN	array of two values used for character values - reading in, temporary storage, and writing out
NARCS	ARCDAT	the number of arcs required for the network representation of the user's data
NBSE	SECOMP	the number of base support equipment resources in the user's data file
NDEC	SENSIT	the number of decision change points located during cost and/or MTBF sensitivity analysis; the number of data points in a set of records on files 17, 18, and 19 (files for sensitivity results)
NDSE	SECOMP	the number of depot support equipment resources in the user's data file

Variable	COMMON or SUBR.	Definition
MENREQ	FMCMP SRUCMP	the *minimum* number of trained maintenance men required to accomplish the expected monthly workload of LRU or SRU repairs
MFB, MFD	MSDAT	maintenance personnel turnover factor, for base and depot, used in computing training requirements and costs
MTBCT	LCOMP	mean end-item operating time between corrective repair tasks
MTBF (MAXLRU)	LRUDAT	an array of real values, one for each LRU, each of which is the mean time between failures for an LRU in its operational environment (LRU operating hours/failure, LRU Data Record)
MTBFHI	MTBFSA	a real value for the high end of an *MTBF sensitivity investigation* range
MTBFIN	MTBFSA	the input value, or baseline, from which MTBF sensitivity analysis is done for an LRU; a value from the MTBF array
MTBFLO, MTBFMI	MTBFSA	real values for the low end and middle of an MTBF sensitivity investigation range
N13, N14	TEMPIN	record location pointers for writing to and reading from the random acess files 13 and 14; files used as push down stacks for data during sensitivity analysis
N15, N16	MAIN	record location pointers for writing to and reading from the random access files 15 and 16; files used for temporary storage of LRU and SRU computed logistics costs

Variable	COMMON or SUBR.	Definition
LWUC (MAXLRU,2)	LRUDAT	an array of values, one double-word entry for each LRU, which are the user assigned alphanumeric LRU identifiers (dimensionless, LRU Data Record)
M	WSDAT	number of bases - the total number of operational locations for the system; each is assumed to have intermediate level repair facilities (bases, Weapon System Data)
MAXARC	N/A	an integer equal to the dimensioned size of arrays for network arc data; used to prevent array overflow
MAXITM	N/A	an integer equal to the dimensioned size of the array for item to SE relationships, ITMSEN; used to prevent array overflow
MAXLFM	N/A	an integer equal to the dimensioned size of arrays for LRU failure mode data; used to prevent array overflow
MAXLRU	N/A	an integer equal to the dimensioned size of arrays for LRU data; used to prevent array overflow
MAXNOD	N/A	an integer equal to the dimensioned size of arrays for network node data; used to prevent array overflow
MAXREF	N/A	an integer equal to the dimensioned size of the arrays for the SE to items relationships, SEXREF and NXTITM; used to prevent array overflow
MAXSE	N/A	an integer equal to the dimensioned size of arrays for SE data; used to prevent array overflow
MAXSRU	N/A	an integer equal to the dimensioned size of arrays for SRU data; used to prevent array overflow

Variable	COMMON or SUBR.	Definition
LLAST	MAIN	integer subscript value used in writing out the LRU and SRU to SE cross reference table; used with LFRST
LNODE	NODPTR	node number for the last node in the network
LOCAP (MAXARC)	SENSIT	an array of values, one for each network arc, used during sensitivity analysis to save the arc capacity values for the low end of each sensitivity investigation interval
LOCAT (15)	SENSIT	an array of character constants, defined in the BLOCK DATA subroutine, used in various WRITE statements
LOFLOW (MAXARC)	SENSIT	an array of values, one for each network arc, used during sensitivity analysis to save the arc flow values for the low end of each sensitivity investigation interval
LOSTAT (MAXNOD)	SENSIT	an array of values, one for each network node, used during sensitivity analysis to save the node state values for the low end of each sensitivity investigation interval
LPTR	N/A	used in numerous places as a pointer to an LRU data record
LRUNAM (MAXLRU,3)	LRUDAT	an array of values, one triple-word entry for each LRU, which are the user assigned LRU alphanumeric descriptors (dimensionless, LRU Data Record)
LRUPTR (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is a pointer to the LRU failure mode data associated with the SRU
LSARC (MAXLFM)	ARCPTR	an array of values, one for each LRU failure mode, each of which is a pointer to the network arc for scrap option costs

Variable	COMMON or SUBR.	Definition
LDSE	NODPTR	node number for the network node which is the last (highest numbered) depot-side SE node
LDSRU	NODPTR	node number for the network node which is the last (highest numbered) depot-side SRU node
LFMOB (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is 0 or is 1 when the user wants to override (prevent) a base decision
LFMOD (MAXLFM)	LFMDAT	same as LFMOB except to override a depot decision
LFMOS (MAXLFM)	LFMDAT	same as LFMOB except to override a scrap decision
LFMS	LFMDAT	counter for the number of LRU failure mode records in the input data file
LFMWUC (MAXLFM,2)	LFMDAT	an array of values, one double-word entry for each LRU failure mode, which are the user assigned alphanumeric LRU identifiers (dimensionless, LRU Failure Mode Data Record)
LFRST	MAIN	integer subscript value used in writing out the LRU and SRU to SE cross reference table; used with LLAST
LINE (MAXSE)	SEIN	an array of values, one for each SE, used for temporary storage of values before writing them
LINES	N/A	counter for the number of output lines written since the previous printer page eject

Variable	COMMON or SUBR.	Definition
L	N/A	same as I
L10 through L5000	N/A	same as I10 through I5000
LASTFM (MAXLRU)	LRUDAT	an array of values, one for each LRU, each of which points to the last set of failure mode data associated with an LRU
LBARC (MAXLFM)	ARCPTR	an array of values, one for each LRU failure mode, each of which is a pointer to the network arc for base repair costs
LBLRU	NODPTR	node number for the network node which is the last (highest numbered) base-side LRU failure mode node
LBSE	NODPTR	node number for the network node which is the last (highest numbered) base-side SE node
LBSRU	NODPTR	node number for the network node which is the last (highest numbered) base-side SRU node
LCRCHI, LCRCLO, & LCRCMI	MTBFSA, UCLSA	during sensitivity analysis the life cycle repair cost associated with the high, low, and middle decision evaluation points
LCSECH, LCSECL, & LCSECM	MTBFSA	during MTBF sensitivity analysis the life cycle support equipment cost associated with the high, low, and middle decision evaluation points
LDARC (MAXLFM)	ARCPTR	an array of values, one for each LRU failure mode, each of which is a pointer to the network arc for depot repair costs
LDLRU	NODPTR	node number for the network node which is the last (highest numbered) depot-side LRU failure mode node

Variable	COMMON or SUBR.	Definition
IND, IND1 *through* IND6	N/A	integer values used as "yes-no" indicators for various conditions in numerous locations, particularly, in the sensitivity analysis subroutines
INDSA	MAIN	integer value, zero through four, indicating sensitivity analysis requirements
INDSAT	SENSIT	integer value, zero or one, indicating sensitivity analysis at extremes only or continuous
IPREV	MAXFLO	node number pointer for backward scanning along an identified augmentation path
ISUB	N/A	integer subscript variable
ITEM	N/A	integer subscript variable for LRUs and SRUs
ITMSEN (MAXITM)	SEXDAT	array of values used for storing the SE reference numbers associated with LRUs and SRUs
ITYPE	SECMP	first character of an SE reference number; used to distinguish base versus depot SE
J	N/A	same as I
J10 through J5000	N/A	same function as I10 through I5000
JUMBO	ARCDAT	very large "cost" value, 2**30, used for the capacity of relational network arcs
K	N/A	same as I
K10 through K5000	N/A	same function as I10 through I5000

Variable	COMMON or SUBR.	Definition
FWDSP (MAXNOD)	NODDAT	an array of values, one for each network node, each of which is a pointer to the first set of arc data with a specific node number as the source node; FWDSP (j) is a pointer to the first arc with node "j" as its source
HICAP (MAXARC)	SENSIT	an array of values, one for each network arc, used during sensitivity analysis to save the arc capacity values for the high end of the sensitivity investigation range
HIFLOW (MAXARC)	SENSIT	an array of values, one for each network arc, used during sensitivity analysis to save the arc flow values for the high end of the sensitivity investigation range
HISTAT (MAXNOD)	SENSIT	an array of values, one for each network node, used during sensitivity analysis to save the node state values for the high end of the sensitivity investigation range
I	N/A	general purpose integer variable for values of transitory significance
I1 through I8	MAIN	temporary storage locations for integer values read from each SRU data card; values are transferred to the proper permanent arrays after the correct subscript is determined
I10 through I5000	N/A	integer values for DO loop parameters, pointers, and subscripts; in general, the value "In" is associated with "DO n" or with "n FORMAT"
ICC	TEMPIN	integer value for printer carriage control
IFLAG	MAIN	integer count of errors detected during input data processing
IMC	SSDAT	initial management cost to introduce a new item (assembly or piece-part) into the Air Force wholesale level inventory system ($/item, Supply System Data Record)

Variable	COMMON or SUBR.	Definition
SBRCT (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the SRU base repair cycle time; the total elapsed time from removal of a failed SRU from an LRU, through intermediate level repair, until it is returned to serviceable base stock (months, SRU Data Record)
SBRCTL	SCOMP	the number of spare SRUs to be purchased to satisfy SRU demands expected to occur during the SRU base repair cycle time
SBRCTP	SCOMP	the expected number of unserviceable SRU assets in the base repair pipeline
SBSARC (MAXSRU)	ARCPTR	an array of values, one for each SRU, each of which is a pointer to the network arc for costs incurred if the next higher assembly LRU is base repaired and the SRU is scrapped or depot repaired
SCRPLC (MAXLFM)	FMCOMP	an array of values, one for each LRU failure mode, each of which is the logistics cost associated with discarding the LRU when it fails
SCRPSC (MAXSRU)	SCOMP	an array of values, one for each SRU, each of which is the logistics cost associated with discarding the SRU when it fails
SDARC (MAXSRU)	ARCPTR	an array of values, one for each SRU, each of which is a pointer to the SRU depot repair cost arc
SDRCTC (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the SRU depot repair cycle time for CONUS bases; the elapsed time from removal of a failed SRU (from the LRU) at a CONUS base until the item could become a serviceable spare in depot stock (months, SRU Data Record)

Variable	COMMON or SUBR.	Definition
SDRCTL	SRUCMP	the number of spare SRUs to be purchased to satisfy SRU demands expected to occur during the SRU depot repair cycle time
SDRCTO (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the SRU depot repair cycle time for overseas bases; the elapsed time from removal of a failed SRU from the LRU at an overseas base until the item could become a serviceable spare in depot stock (months, SRU Data Record)
SDRCTP	SRUCMP	the expected number of unserviceable SRU assets in the depot repair pipeline
SEACQ	SECMP	*support equipment acquisition* cost for a particular type of SE
SEARCP (MAXSE)	SECOMP	an array of values, one for each SE, each of which is a pointer to the SE cost arc of the network
SECODE (MAXSE)	SECOMP	an array of values, one for each SE, each of which is a user assigned 4 digit integer value, from 1000 to 8999, used as an SE identifier
SECOST (MAXSE)	SECOMP	an array of values, one for each SE, each of which is the total facilities, acquisition, and operations cost for the SE
SEFAC	SECMP	cost of new *facilities* for support equipment
SEOPNS	SECMP	life cycle operations and maintenance cost for a support equipment
SEPF (MAXSE)	SECOMP	an array of values, one for each SE, each of which is a pointer to the first entry in the linked list of items (LRUs and SRUs) requiring the SE; the linked list is in arrays SEXREF and NXTITM

Variable	COMMON or SUBR.	Definition
SEPL (MAXSE)	SECOMP	an array of values, one for each SE, each of which is a pointer to the last entry in the linked list of items requiring the SE (see SEPF)
SEPTFM (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is a pointer to the last value in ITMSEN associated with the failure mode
SEPTL (MAXLRU)	LRUDAT	an array of values, one for each LRU, each of which is a pointer to the last value in ITMSEN associated with the LRU
SEPTR	N/A	variable used at various places as a pointer to SE data
SEPTS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is a pointer to the last value in ITMSEN associated with the SRU
SERN (MAXSE + 16)	TEMPIN	an array of values used for temporary storage of SE resource numbers
SEUR (MAXSE)	SECOMP	an array of values, one for each SE, each of which is the utilization rate, e.g., 80 percent, for the SE
SEXREF (MAXREF)	SEXDAT	an array of values each of which is a pointer to an LRU failure mode or an SRU; it is used with the array NXTITM to give a support equipment to items cross reference list
SFHI, SFLO	MAIN	input values specifying the high and low end ratios for sensitivity analysis investigations (dimensionless, Weapon System Data Record)
SFMS	SRUDAT	counter for the number of SRUs identified on LRU failure mode data cards

Variable	COMMON or SUBR.	Definition
SMTBFH SMTBFL	MAIN	the high end and low end range values used for sensitivity analysis of an MTBF
SOB (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is 0 or is 1 if the user wants to override (prevent) a base repair decision for the SRU (dimensionless, SRU Data Record)
SOD (MAXSRU)	SRUDAT	same as SOB except it is for the depot decision
SOS (MAXSRU)	SRUDAT	same as SOB except it is for the scrap decision
SOSTPL	SRUCMP	the expected number of serviceable SRU assets in the depot to base order and shipping time pipeline
SOSTSL	SRUCMP	the number of spare SRUs to be purchased to satisfy SRU demands expected to occur during an order and shipping time cycle
SPTR	N/A	general purpose pointer to SRU data
SPTRSA	MAIN	SRU pointer from sensitivity analysis results
SRC	SSDAT	cost for shipping items to CONUS locations ($/pound, Supply System Data Record)
SRCE (MAXARC)	ARCDAT	an array of values, one for each network arc, each of which is the node number of the source node
SRO	SSDAT	cost for shipping items to overseas locations ($/pound, Supply System Data Record)

Variable	COMMON or SUBR.	Definition
SRUNAM (MAXSRU, 2)	SRUDAT	an array of values, one double-word entry for each SRU, which are the user assigned SRU alphanumeric descriptors (dimensionless, LRU Failure Mode Data Record)
SRUPTR (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is zero or is a pointer to the SRU data associated with the failure mode
SSARC (MAXSRU)	ARCPTR	an array of values, one for each SRU, each of which is a pointer to the network arc for scrap option costs
SSEHR (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the number of support equipment hours required for repair of an SRU (hours/repair, SRU data record)
SSEUHB (MAXSRU)	SCOMP	an array of values, one for each SRU, each of which is the SRU's SE utilization hours per month at base level
SSEUHD (MAXSRU)	SCOMP	an array of values, one for each SRU, each of which is the SRU's SE utilization hours per month at depot level
STATE (MAXNOD)	NODDAT	an array of values, one for each network node, each of which is used to record the labelling process state values during the max-flow determination
SWDRCT	SRUCMP	SRU weighted depot repair cycle time
SWUC (MAXSRU, 2)	SRUDAT	an array of values, one double-word entry for each SRU, which are the user assigned alphanumeric SRU identifiers (dimensionless, LRU Failure Mode Data Record and SRU Data Record)
SYSNAM (3)	WSDAT	an array of 3 values specifying an alphanumeric description for the end-item (dimensionless, Weapon System Data Record)
TD	SSDAT	cost per original page of technical data produced by the contractor to support item repairs ($/page, Supply Data Record)

Variable	COMMON or SUBR.	Definition
TDPLFM (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is the number of technical data pages required to perform an LRU repair task (No. pages, LRU Failure Mode Data Record) NOTE: in Chapter 3 of the User's Guide this variable is NTDFM
TDPS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the number of technical data pages required for SRU repairs (No. pages, SRU Data Record) NOTE: in Chapter 3 of the User's Guide this variable is NTDS
TFAILP (MAXLRU)	LCOMP	an array of values, one for each LRU, each of which is the sum of the FAILP values for the failure modes of the LRU
TLCD	LCOMP	total life cycle repair demands for an LRU at each base
TLCDF	FMCOMP	total life cycle repair demands for a particular failure mode of an LRU at each base
TQCTGM	LCOMP	total questionable corrective tasks generated monthly at each base
TRB	MSDAT	annual turnover rate for intermediate level maintenance personnel (fraction of personnel replaced/year, Maintenance System Data Record)
TRCLFM (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is the expected training cost, instruction and materials, for LRU repairs ($/man/week, LRU Failure Mode Data Record) NOTE: in Chapter 3 of the User's Guide this variable is TRC

Variable	COMMON or SUBR.	Definition
TRCS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the expected training cost, instruction and materials, for SRU repairs ($/man/week, SRU Data Record)
TRD	MSDAT	annual turnover rate for depot level maintenance personnel (fraction of personnel replaced/year, Maintenance System Data Record)
TRWLFM (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is the amount of training time required for LRU repairs (weeks, LRU Failure Mode Data Record) NOTE: in Chapter 3 of the User's Guide this variable is TRW
TRWS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the amount of training time required for SRU repairs (weeks, SRU Data Record)
TSCRAP	LCOMP	total life cycle cost for the LRU scrap option
UCL (MAXLRU)	LRUDAT	an array of values, one for each LRU, each of which is the unit cost of the LRU ($, LRU Data Record)
UCLIN, UCLSH, UCLSL	MAIN	LRU unit cost values used for sensitivity analysis - input value, sensitivity high value, and sensitivity low value
UCPPFM (MAXLFM)	LFMDAT	an array of values, one for each LRU failure mode, each of which is the total cost of all non-reparable assemblies and/or piece-parts required for repair of an LRU failure mode ($, LRU Failure Mode Data Record)
UCPPS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the total cost of all non-reparable assemblies and/or piece-parts required for repair of an SRU ($, SRU Data Record)

Variable	COMMON or SUBR.	Definition
UCS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the unit cost of an SRU, ($/SRU, SRU Data Record)
UCSIN, UCSSH, UCSSL	MAIN	SRU unit cost values used for sensitivity analysis - input value, sensitivity high value, and sensitivity low value
UEBASE	WSDAT	the number of end-items operating at each base (dimensionless, Weapon System Data Record)
UF (MAXLRU)	LRUDAT	an array of values, one for each LRU, each of which is an LRU utilization factor; ratio of LRU operating hours to end-item operating hours (dimensionless, LRU Data Record)
UR	WSDAT	end-item utilization rate (operating hours/month, Weapon System Data Record)
USEHRS (MAXSE)	SECOMP	an array of values, one for each SE, each of which is the total usage hours required on the SE, per month, for all LRUs and SRUs
WCFF	WCF	wholesale change factor for LRU MTBF (dimensionless, wholesale changes data card)
WCFSE	WCF	wholesale change factor for support equipment cost (dimensionless, wholesale changes data card)
WCFUC	WCF	wholesale change factor for LRU, SRU, and pieceparts cost (dimensionless, wholesale changes data card)
WDRCT	LCOMP	weighted average of CONUS and overseas depot repair cycle times for an LRU

Variable	COMMON or SUBR.	Definition
WGTL (MAXLRU)	LRUDAT	an array of values, one for each LRU, each of which is the weight of the LRU (pounds, LRU Data Record)
WGTS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the weight of the SRU (pounds, SRU Data Record)
WOST	SSDAT	weighted average of CONUS and overseas order and shipping times
WPSR	SSDAT	weighted average of CONUS and overseas packing and shipping rates
WTPPFM	LRUDAT	an array of values, one for each LRU failure mode, each of which is the weight of non-reparable assemblies and/or piece-parts required of LRU repair (pounds, LRU Failure Mode Data Record)
WTPPS (MAXSRU)	SRUDAT	an array of values, one for each SRU, each of which is the weight of non-reparable assemblies and/or piece-parts required for SRU repair (pounds, SRU Data Record)
XSE (MAXSE,3)	SEIN	an array of values, one triple-word entry for each SE, which are alphanumeric descriptors for the SE resources (dimensionless, Support Equipment Data Record)

APPENDIX B. NRLA Program Array Dimensions

Introduction. The purpose of this Appendix is to provide detailed information about the dimensioned sizes of arrays in the program. The program user needs this information so that he can determine if the default array sizes are adequate for his data file or if some tailoring of array sizes is necessary.

The arrays of the program can be grouped according to the type of data they hold. One such classification is shown in Table B.1 with the number of arrays of each type and the default size for each array. As the Table indicates, there are 16 arrays for support equipment data and they can hold information for up to 20 support equipment resources. Similarly, there are 16 arrays for LRU data which are dimensioned for up to 25 different LRUs. The adequacy of the default sizes can be determined for some classifications by simply counting input data records (e.g., support equipment and LRUs). For other classifications, network nodes and network arcs, a simple formula is used to determine the array size requirements.

Where the default sizes are insufficient, program changes are required. These changes must be made in the main routine of the program and in every subroutine (except SORT). Although the program changes are straightforward and relatively simple, they must be done accurately. Errors committed while redimensioning arrays could cause program aborts or, even worse, could cause computational errors which could remain undetected.

The remaining paragraphs of this Appendix give details about the dimensioning statements for the array classes listed in Table B.1. A figure is provided for each classification showing the program's dimensioning statements with their default sizes and with increased sizes. The Appendix concludes with a discussion of data file changes required as a function of redimensioning arrays in the program.

Support Equipment. Arrays for support equipment data are contained in the labelled common areas SEIN and SECOMP as shown in Figure B.1a. The default size for these arrays is 20. For data files containing more than 20 SE resources, these 16 arrays must be increased in size to accommodate the input data. In addition, the value for MAXSE must be changed so that it is equal to the new dimensioned size for the arrays. Figure B.1b shows the arrays dimensioned for up to 35 SE resources and MAXSE changed accordingly.

LRUs. As LRU factors are read, from type '31' data records, they are stored in the arrays shown in Figure B.2a. These 16 arrays need to be dimensioned at least as large as the number of LRU data records in the input file. Figure B.2b shows the arrays changed to allow up to 40 LRU data records and the value for MAXLRU changed to specify the new maximum sizes.

LRU Failure Modes. The arrays for LRU failure mode input data, from type '41' data records, and the arrays for computed values are defined in the labelled COMMON areas LFMDAT and FMCOMP, respectively. These 27 arrays are shown in Figure B.3a with their default dimensions. Increased array sizes allowing up to 90 failure modes are shown in Figure B.3b with the appropriate change to MAXLFM.
NOTE: There are four other arrays for LRU failure mode related data which must be dimensioned the same as the arrays of Figure B.3. Three of these are listed with Item to Arc Pointers and the other with Sensitivity Analysis arrays.

TABLE B.1. NRLA Program Array Classifications

Array Classification	No. of Arrays	Default Size
Support Equipment	16	20
LRUs	16	25
LRU Failure Modes	27	40
SRUs	33	40
Item-to-Arc Pointers	8	40
SE Cross Reference	3	210
Network Nodes	6	200
Network Arcs	7	400
Sensitivity Analysis	13	mixture

SRUs. The 33 arrays defined for SRU input values and for SRU computed values are shown in Figure B.4a with their default sizes. Figure B.4b shows the arrays redimensioned to accommodate up to 80 SRU data records, type '51' records, in the input file.
NOTE: There are six other arrays for SRU related data which need to be dimensioned the same as the arrays of Figure B.4. Five of these are listed with Item to Arc Pointers and the other with Sensitivity Analysis arrays.

Item to Arc Pointers. The 8 array in the COMMON area ARCPTR provide storage space for pointers from item data to the corresponding network arc data. The arrays LDARC, LSARC, and LBARC contain the LRU failure mode data to network arc pointers. These 3 arrays need to be dimensioned the same as the other failure mode related arrays. The 5 remaining arrays contain pointers from SRU data to the corresponding network arc data values. Figure B.5a shows the arrays with their default sizes and Figure B.5b shows them modified to be consistent with MAXFLM and MAXSRU of Figure B.3 and B.4, respectively.

SE Cross Reference. The 3 arrays of the labelled COMMON area SEXDAT contain the item-to-SE requirements data and that data inverted into SE-to-item relationships.

As SE requirements data is read from the type '32', '42', and '52' data records, it is saved sequentially in the array ITMSEN. Consequently, the size requirement for this array can be determined by counting the SE Resource Numbers specified on the '32', '42', and '52' records. The dimensioned size for ITMSEN must be assigned as the value for the variable MAXITM.

The arrays SEXREF and NXTITM are used for the SE-to-item relationships. In general, these arrays need to be larger than ITMSEN. This occurs because an SE Resource Number from a type '32' card (LRU requirements) is stored once in ITMSEN, but it requires a separate entry in SEXREF and NXTITM for every failure mode of the LRU. Thus, if a '32' card has 4 SE Resource Numbers, it will use 4 entries in ITMSEN. Then, if the corresponding LRU has 10 failure modes, it will need 40 entries of SEXREF and ITMSEN. Similarly, if the LRU has 20 failure modes, it will need 80 entries of SEXREF and NXTITM. Therefore, determining the array size requirements for SEXREF and NXTITM is a two part computation.

```
01031 C      TEMPORARY DATA INPUT AREAS
01032       COMMON /TEMPIN/ICC,CARD,WUC,C1,C2,C12,C2PREV,SEN,SERN,NAME,
01033      &    CHARS,N13,N14,IMON,IDAY,IYR,IPAGE
01034       INTEGER CARD(20),WUC(2),NAME(2),CHARS(15)
01035       INTEGER C1,C2,C12,C2PREV,SEN,SERN(36)

01056 C      TYPE & DIMENSION STATEMENTS FOR SUPPORT EQUIPMENT (SE) DATA
01057 C          SUPPORT EQUIPMENT INPUT VALUES
01058       COMMON /SEIN/XSE,LINE,CADB,TECHDP,TRNGW,TRNGC,CODB,FDB,
01059      &               OPHRS,NSECI,BSYHRS
01060       INTEGER XSE(20,3),LINE(20)
01061       DIMENSION CADB(20),CODB(20)
01062       DIMENSION FDB(20),OPHRS(20),NSECI(20),BSYHRS(20)
01063 C          COMPUTED SE VALUES
01064       COMMON /SECOMP/SECODE,SEPF,SEPL,REQMT,USEHRS,SEUR,SECOST,
01065      &                 SEARCP,NUMSER,NDSE,NBSE
01066       INTEGER SECODE(20),SEPF(20),SEPL(20),REQMT(20),SEARCP(20)
01067       DIMENSION USEHRS(20),SEUR(20),SECOST(20)
01068       DATA MAXSE/20/
```

Figure B.1a. Support equipment arrays with default sizes.

```
01031 C      TEMPORARY DATA INPUT AREAS
01032       COMMON /TEMPIN/ICC,CARD,WUC,C1,C2,C12,C2PREV,SEN,SERN,NAME,
01033      &    CHARS,N13,N14,IMON,IDAY,IYR,IPAGE
01034       INTEGER CARD(20),WUC(2),NAME(2),CHARS(15)
01035       INTEGER C1,C2,C12,C2PREV,SEN,SERN(51)

01056 C      TYPE & DIMENSION STATEMENTS FOR SUPPORT EQUIPMENT (SE) DATA
01057 C          SUPPORT EQUIPMENT INPUT VALUES
01058       COMMON /SEIN/XSE,LINE,CADB,TECHDP,TRNGW,TRNGC,CODB,FDB,
01059      &               OPHRS,NSECI,BSYHRS
01060       INTEGER XSE(35,3),LINE(35)
01061       DIMENSION CADB(35),CODB(35)
01062       DIMENSION FDB(35),OPHRS(35),NSECI(35),BSYHRS(35)
01063 C          COMPUTED SE VALUES
01064       COMMON /SECOMP/SECODE,SEPF,SEPL,REQMT,USEHRS,SEUR,SECOST,
01065      &                 SEARCP,NUMSER,NDSE,NBSE
01066       INTEGER SECODE(35),SEPF(35),SEPL(35),REQMT(35),SEARCP(35)
01067       DIMENSION USEHRS(35),SEUR(35),SECOST(35)
01068       DATA MAXSE/35/
```

Figure B.1b. Support equipment arrays with increased sizes.

Figure B.1. Support equipment arrays.

```
C      ARRAYS FOR LRU DATA ( DATA FOR ALL FAILURE MODES )
C          LRU DATA INPUT VALUES
      COMMON /LRUDAT/NLRU,LWUC,UCL,MTBF,RIP,UF,QPA,WGTL,DRCTC,DRCTO,
     &                BRCT,LRUNAM,NSERL,SEPTL,FRSTFM,LASTFM
C
      INTEGER LWUC(25,2),LRUNAM(25,3)
      REAL MTBF
      INTEGER FRSTFM(25),LASTFM(25),SEPTL(25)
      DIMENSION UCL(25),MTBF(25),RIP(25),UF(25),QPA(25),WGTL(25)
      DIMENSION DRCTC(25),DRCTO(25),BRCT(25),NSERL(25),TFAILP(25)
      DATA MAXLRU/25/
```

Figure B.2a. LRU arrays with default sizes.

```
C      ARRAYS FOR LRU DATA ( DATA FOR ALL FAILURE MODES )
C          LRU DATA INPUT VALUES
      COMMON /LRUDAT/NLRU,LWUC,UCL,MTBF,RIP,UF,QPA,WGTL,DRCTC,DRCTO,
     &                BRCT,LRUNAM,NSERL,SEPTL,FRSTFM,LASTFM
C
      INTEGER LWUC(40,2),LRUNAM(40,3)
      REAL MTBF
      INTEGER FRSTFM(40),LASTFM(40),SEPTL(40)
      DIMENSION UCL(40),MTBF(40),RIP(40),UF(40),QPA(40),WGTL(40)
      DIMENSION DRCTC(40),DRCTO(40),BRCT(40),NSERL(40),TFAILP(40)
      DATA MAXLRU/40/
```

Figure B.2b. LRU arrays with increased sizes.

Figure B.2 LRU arrays.

```
C
C      ARRAYS FOR LRU FAILURE MODE DATA
C          FAILURE MODE DATA INPUT VALUES
      COMMON /LFMDAT/LFMS,LFMWUC,FMNUM,FAILP,UCPPFM,WTPPFM,NPPAFM,
     &        FMSEHR,NABFM,DMMHFM,BMMHFM,TDPLFM,TRWLFM,TRCLFM,NTDLFM,
     &        NTBLFM,NSERFM,LFMOD,LFMOS,LFMOB,SEPTFM,NFMS,SRUPTR,OPDECL
      INTEGER LFMWUC(40,2),OPDECL(40)
      DIMENSION FAILP(40),UCPPFM(40),WTPPFM(40),NPPAFM(40),FMSEHR(40)
      DIMENSION NABFM(40),DMMHFM(40),BMMHFM(40),TDPLFM(40)
      DIMENSION TRWLFM(40),TRCLFM(40),NTDLFM(40),NTBLFM(40)
      DIMENSION NSERFM(40),LFMOD(40),LFMOS(40),LFMOB(40)
      INTEGER SRUPTR(40),SEPTFM(40),FMNUM(40)
      DATA MAXLFM/40/
C          QUANTITIES COMPUTED BY SUBROUTINE FMCMP
      COMMON/FMCOMP/DEPOLC,SCRPLC,BASELC,TLCDF,FSEUHD,FSEUHB
      DIMENSION DEPOLC(40),SCRPLC(40),BASELC(40),FSEUHD(40),FSEUHB(40)
```

Figure B.3a. LRU failure mode arrays with default sizes.

```
C
C      ARRAYS FOR LRU FAILURE MODE DATA
C          FAILURE MODE DATA INPUT VALUES
      COMMON /LFMDAT/LFMS,LFMWUC,FMNUM,FAILP,UCPPFM,WTPPFM,NPPAFM,
     &        FMSEHR,NABFM,DMMHFM,BMMHFM,TDPLFM,TRWLFM,TRCLFM,NTDLFM,
     &        NTBLFM,NSERFM,LFMOD,LFMOS,LFMOB,SEPTFM,NFMS,SRUPTR,OPDECL
      INTEGER LFMWUC(90,2),OPDECL(90)
      DIMENSION FAILP(90),UCPPFM(90),WTPPFM(90),NPPAFM(90),FMSEHR(90)
      DIMENSION NABFM(90),DMMHFM(90),BMMHFM(90),TDPLFM(90)
      DIMENSION TRWLFM(90),TRCLFM(90),NTDLFM(90),NTBLFM(90)
      DIMENSION NSERFM(90),LFMOD(90),LFMOS(90),LFMOB(90)
      INTEGER SRUPTR(90),SEPTFM(90),FMNUM(90)
      DATA MAXLFM/90/
C          QUANTITIES COMPUTED BY SUBROUTINE FMCMP
      COMMON/FMCOMP/DEPOLC,SCRPLC,BASELC,TLCDF,FSEUHD,FSEUHB
      DIMENSION DEPOLC(90),SCRPLC(90),BASELC(90),FSEUHD(90),FSEUHB(90)
```

Figure B.3b. LRU failure mode arrays with increased sizes.

Figure B.3 LRU failure mode arrays.

```
01710 C
01720 C      ARRAYS FOR SRU DATA
01730 C          SRU DATA INPUT VALUES
01740       COMMON /SRUDAT/SFMS,SWUC,SRUNAM,UCS,UCPPS,WGTS,WTPPS,SDRCTC,
01750      &         SSEHR,SDRCTO,SBRCT,NPPAS,NABS,DMMHS,BMMHS,TDPS,TRWS,
01760      &        TRCS,NTDS,NTBS,NSERS,SOD,SOS,SOB,SEPTS,NSRUS,LRUPTR,OPDECS
01770       INTEGER SWUC(40,2),SRUNAM(40,2)
01780       INTEGER OPDECS(40)
01790       INTEGER SFMS,SOD(40),SOS(40),SOB(40),SEPTS(40)
01800       DIMENSION UCS(40),UCPPS(40),WGTS(40),WTPPS(40),SDRCTC(40)
01810       DIMENSION SDRCTO(40),SBRCT(40),NPPAS(40),NABS(40),SSEHR(40)
01820       DIMENSION DMMHS(40),BMMHS(40),TDPS(40),TRWS(40),TRCS(40)
01830       DIMENSION NTDS(40),NTBS(40),NSERS(40),LRUPTR(40)
01840       DATA MAXSRU/40/
01850 C          QUANTITIES COMPUTED BY SUBROUTINE SRUCMP
01860       COMMON /SCOMP/DEPOSC,SCRPSC,BASESC,BRLDS,BRLSS,SBRCTP,SBRCTL,
01870      &        SSEUHD,SSEUHB
01880       DIMENSION DEPOSC(40),SCRPSC(40),BASESC(40),BRLDS(40),BRLSS(40)
01890       DIMENSION SSEUHD(40),SSEUHB(40)
```

Figure B.4a. SRU array with default sizes.

```
01710 C
01720 C      ARRAYS FOR SRU DATA
01730 C          SRU DATA INPUT VALUES
01740       COMMON /SRUDAT/SFMS,SWUC,SRUNAM,UCS,UCPPS,WGTS,WTPPS,SDRCTC,
01750      &         SSEHR,SDRCTO,SBRCT,NPPAS,NABS,DMMHS,BMMHS,TDPS,TRWS,
01760      &        TRCS,NTDS,NTBS,NSERS,SOD,SOS,SOB,SEPTS,NSRUS,LRUPTR,OPDECS
01770       INTEGER SWUC(80,2),SRUNAM(80,2)
01780       INTEGER OPDECS(80)
01790       INTEGER SFMS,SOD(80),SOS(80),SOB(80),SEPTS(80)
01800       DIMENSION UCS(80),UCPPS(80),WGTS(80),WTPPS(80),SDRCTC(80)
01810       DIMENSION SDRCTO(80),SBRCT(80),NPPAS(80),NABS(80),SSEHR(80)
01820       DIMENSION DMMHS(80),BMMHS(80),TDPS(80),TRWS(80),TRCS(80)
01830       DIMENSION NTDS(80),NTBS(80),NSERS(80),LRUPTR(80)
01840       DATA MAXSRU/80/
01850 C          QUANTITIES COMPUTED BY SUBROUTINE SRUCMP
01860       COMMON /SCOMP/DEPOSC,SCRPSC,BASESC,BRLDS,BRLSS,SBRCTP,SBRCTL,
01870      &        SSEUHD,SSEUHB
01880       DIMENSION DEPOSC(80),SCRPSC(80),BASESC(80),BRLDS(80),BRLSS(80)
01890       DIMENSION SSEUHD(80),SSEUHB(80)
```

Figure B.4b. SRU array with increased sizes.

Figure B.4. SRU arrays.

```
02190 C
02200       COMMON /ARCPTR/LDARC,LSARC,LBARC,SDARC,SSARC,SBARC,
02210      &                SBSARC,SBDARC
02220       INTEGER LDARC(40),LSARC(40),LBARC(40)
02230       INTEGER SDARC(40),SSARC(40),SBARC(40),SBSARC(40),SBDARC(40)
```

Figure B.5a. Item to arc pointer arrays with default sizes.

```
02190 C
02200       COMMON /ARCPTR/LDARC,LSARC,LBARC,SDARC,SSARC,SBARC,
02210      &                SBSARC,SBDARC
02220       INTEGER LDARC(90),LSARC(90),LBARC(90)
02230       INTEGER SDARC(80),SSARC(80),SBARC(80),SBSARC(80),SBDARC(80)
```

Figure B.5b. Item to arc pointer arrays with increased sizes.

Figure B.5. Item to arc pointer arrays.

```
01910 C
01920 C      ARRAYS FOR SE CROSS REFERENCE
01930       COMMON /SEXDAT/SEXREF,NXTITM,ITMSEN
01940       INTEGER SEXREF(210),NXTITM(210),ITMSEN(210)
01950       DATA MAXREF,MAXITM/210,210/
```

Figure B.6a. SE cross reference arrays with default sizes.

```
01910 C
01920 C      ARRAYS FOR SE CROSS REFERENCE
01930       COMMON /SEXDAT/SEXREF,NXTITM,ITMSEN
01940       INTEGER SEXREF(500),NXTITM(500),ITMSEN(250)
01950       DATA MAXREF,MAXITM/500,250/
```

Figure B.6b. SE cross reference arrays with increased sizes.

Figure B.6. SE cross reference arrays.

The first part uses each type '32' card separately. The number of SE Resource Numbers on the card is multiplied by the number of failure modes identified for the corresponding LRU. The second part of the computation is summing these individual products, plus the number of SE Resource Numbers on type '42' cards, plus the number of SE Resource Numbers on type '52' cards. This sum is the number of SE-to-item relationships.

Figure B.6a shows the SE cross reference arrays with their default dimensions. Changing the dimensions to allow up to 250 entries on '32', '42', and '52' data records and to allow up to 500 SE-to-item relationships is shown in Figure B.6b. Note that MAXREF is set to 500 to correspond to the dimensions for SEXREF and NXTITM and that MAXITM is set to 250 to match the dimension for ITMSEN.

NETWORK NODES. The six arrays for network node information are shown in Figure B.7a with their default size of 200. The size requirement for these arrays is computed using the formula:

2+(No. of SE) +2* (No. of LRU failure modes) +2* (No. of SRUs)

For a data file with 33 SE resource data cards, 85 LRU failure mode data cards, and 75 SRU data cards, the computation would be:

$$2+(33) +2*(85) +2*(75) = 355$$

Figure B.7b shows redimensioning to this size and changing the value of MAXNOD to equal the new size. This computed value, 355, provides precisely the amount of space required and, therefore, has no allowance for additional SE, failure modes, or SRUs which might subsequently be identified. Consequently, it should be dimensioned slightly larger.

An alternate method for determining the array sizes is to use the maximum array sizes for SE, LRU failure modes, and SRUs rather than the actual, or anticipated, number in a data file. Thus, the above formula becomes:

2 +(MAXSE) + 2*(MAXFLM) +2*(MAXSRU)

Using values from Figures B.1b, B.3b, and B.4b, the result is:

$$2+(35) +2*(90) +2*(80) =377$$

NOTE: See the Sensitivity Analysis paragraph for 2 additional arrays of network node data which must be changed when MAXNOD changes.

NETWORK ARCS. Arrays for network arc data values are located in the labelled COMMON area ARCDAT as shown in Figure B.8a with their default dimensions. The size requirement for these arrays is computed using the formula:

(No. of SE) + 3* (No. of LRU failure modes) + 6* (No. of SRUs) + (No. of SE-to-item relationships)

The last term is determined as described previously for the SEXREF and NXTITM arrays. Similar to the network node computation, an alternate form for the formula using maximum array sizes is:

(MAXSE) + 3* (MAXLFM) + 6* (MAXSRU) + (MAXREF)

```
01970 C
01980 C      VECTORS FOR NETWORK SOLUTION DATA
01990       COMMON /NODDAT/NNODES,NPATH,DLTAFL,STATE,OPSTAT,FWDSP,BACKSP
02000       COMMON /NODPTR/LDSE,LDLRU,LDSRU,LBLRU,LBSRU,LBSE,LNODE
02010 C        NODE LABELING DATA
02020       INTEGER NPATH(200),DLTAFL(200),STATE(200)
02030       INTEGER OPSTAT(200)
02040 C        NODE FORWARD & BACKWARD SCAN POINTERS
02050       INTEGER FWDSP(200),BACKSP(200)
02060 C     DATA MAXNOD/200/
```

Figure B.7a. Network node arrays with default sizes.

```
01970 C
01980 C      VECTORS FOR NETWORK SOLUTION DATA
01990       COMMON /NODDAT/NNODES,NPATH,DLTAFL,STATE,OPSTAT,FWDSP,BACKSP
02000       COMMON /NODPTR/LDSE,LDLRU,LDSRU,LBLRU,LBSRU,LBSE,LNODE
02010 C        NODE LABELING DATA
02020       INTEGER NPATH(355),DLTAFL(355),STATE(355)
02030       INTEGER OPSTAT(355)
02040 C        NODE FORWARD & BACKWARD SCAN POINTERS
02050       INTEGER FWDSP(355),BACKSP(355)
02060 C     DATA MAXNOD/355/
```

Figure B.7b. Network node arrays with increased sizes.

Figure B.7. Network node arrays.

```
02070 C       ARC DATA
02080       COMMON /ARCDAT/NARCS,JUMBO,SRCE,DEST,CAP,FLOW,BKPTR,SAVCAP,SAVFLO
02090       INTEGER SRCE(400),DEST(400),CAP(400),FLOW(402),BKPTR(400)
02100       INTEGER SAVCAP(400),SAVFLO(402)
02110       DATA MAXARC/400/
```

Figure B.8a. Network arc arrays with default sizes.

```
02070 C       ARC DATA
02080       COMMON /ARCDAT/NARCS,JUMBO,SRCE,DEST,CAP,FLOW,BKPTR,SAVCAP,SAVFLO
02090       INTEGER SRCE(1270),DEST(1270),CAP(1270),FLOW(1272),BKPTR(1270)
02100       INTEGER SAVCAP(1270),SAVFLO(1272)
02110       DATA MAXARC/1270/
```

Figure B.8b. Network arc arrays with increased sizes.

Figure B.8. Network arc arrays.

```
02120       COMMON/SENSIT/INDSAT,LOSTAT,HISTAT,LOCAP,HICAP,LOFLOW,HIFLOW,
02130      &        NDEC,SADECL,SADECS,DECVLO,DECVHI,NLSCHG,NSECHG,LOCAT
02140       INTEGER LOCAT(15)
02150       INTEGER LOSTAT(200),HISTAT(200)
02160       INTEGER LOCAP(400),HICAP(400),LOFLOW(400),HIFLOW(400)
02170       INTEGER SADECL(40,10),SADECS(40,10)
02180       DIMENSION DECVLO(10),DECVHI(10),NLSCHG(10),NSECHG(10)
```

Figure B.9a. Sensitivity analysis arrays with default sizes.

```
02120       COMMON/SENSIT/INDSAT,LOSTAT,HISTAT,LOCAP,HICAP,LOFLOW,HIFLOW,
02130      &        NDEC,SADECL,SADECS,DECVLO,DECVHI,NLSCHG,NSECHG,LOCAT
02140       INTEGER LOCAT(15)
02150       INTEGER LOSTAT(355),HISTAT(355)
02160       INTEGER LOCAP(1270),HICAP(1270),LOFLOW(1270),HIFLOW(1270)
02170       INTEGER SADECL(90,10),SADECS(80,10)
02180       DIMENSION DECVLO(10),DECVHI(10),NLSCHG(10),NSECHG(10)
```

Figure B.9b. Sensitivity analysis arrays with increased sizes.

Figure B.9. Sensitivity analysis arrays.

Using values from the nodes computations and using 500 for both SE-to-item relationships and MAXREF these formulas give:

$$(33)+3*(85)+6*(75)+(500) = 1238$$

$$(35)+3*(90)+6*(80)+(500) = 1285$$

Figure B.8b shows the arc arrays dimensioned to hold data for up to 1270 network arcs. As shown, the value of MAXARC must be set equal to the dimensioned size. Also, it is mandatory that the arrays FLOW and SAVFLO be dimensioned for 2 values larger than MAXARC. (The program uses the values 'MAXARC+1' and 'MAXARC+2' as subscripts for these arrays.)

SENSITIVITY ANALYSIS. Arrays in the labelled COMMON area SENSIT are used for temporary storage of values by the program's sensitivity analysis subroutines. These arrays are shown Figure B.9a with their default sizes.

Two of these arrays, LOSTAT and HISTAT, must be dimensioned for the maximum number of nodes in the network (MAXNOD), that is, for the same size as the arrays in Figure B.7. The arrays LOCAP, HICAP, LOFLOW, and HIFLOW must be dimensioned for the maximum number of network arcs (MAXARC); see Figure B.8. The arrays SADECL and SADECS hold up to 10 values per LRU failure mode and SRU, respectively. Consequently, the first dimension value for these arrays needs to match MAXLFM and MAXSRU, respectively. Figure B.9b shows the results of changing the default values to be consistent with Figures B.3, B.4, B.7, and B.8. (Note that the arrays LOCAT, DECVLO, DECVHI, NLSCHG, and NSECHG do not need to be redimensioned. This is also true for the arrays CARD, WUC, NAME, CHARS, and SERN in the labelled COMMON area TEMPIN.)

SUMMARY OF DIMENSIONING REQUIREMENTS. As indicated in previous paragraphs, the arrays of the program can be listed in sets such that all arrays in a group must be dimensioned consistently. This grouping is a minor reorganization from the one in Table B.1 and is shown in Table B.2.

The fourth column is labelled MAXLRU to indicate that it is an alphabetical listing of the arrays which must be dimensioned for the maximum number of LRUs in the user's data file. The column contains the same 16 array names shown in Figure B.2.

Similarly, the seventh column lists the arrays for support equipment data. Each of these must be dimensioned for the maximum number of SE resources in the user's data file.

The third column is labelled MAXLFM to indicate that each array must be dimensioned for the maximum number of LRU Failure Modes in the user's data file. The column contains the 26 arrays of Figure B.3, plus 3 arrays from Figure B.5, and 1 array from Figure B.9

All arrays for SRU data are shown in the last column - 32 arrays from Figure B.4, plus 5 arrays from Figure B.5, and 1 from Figure B.9. The columns labelled MAXARC and MAXNOD contain the network arc related arrays and the network node related arrays, respectively.

The two remaining columns, labelled MAXITM and MAXREF, show the arrays for the SE cross reference data.

Table B.2. Summary of Dimensioning Requirements for NRLA Arrays

MAXARC	MAXITM	MAXLFM	MAXLRU	MAXNOD	MAXREF	MAXSE	MAXSRU
BKPTR	ITMSEN	BASELC	BRCT	BACKSP	NXTITM	bSYHRS	BASESC
CAP		BMMHFM	DRCTC	DLTAFL	SEXREF	CADB	BMMHS
DEST		DEPOLC	DRCTO	FWDSP		CODB	BRLDS
FLOW		DMMHFM	FRSTFM	HISTAT		FDB	BRLSS
HICAP		FAILP	LASTFM	LOSTAT		LINE	DEPOSC
HIFLOW		FMNUM	LRUNAM	NPATH		NSECI	DMMHS
LOCAP		FMSEHR	LWUC	OPSTAT		OPHRS	LRUPTR
LOFLOW		FSEUHB	MTBF	STATE		REQMT	NABS
SAVCAP		FSEUHD	NSERL			SEARCP	NPPAS
SAVFLO		LBARC	QPA			SECODE	NSERS
SRCE		LDARC	RIP			SECOST	NTBS
		LFMOB	SEPTL			SEPF	NTDS
		LFMOD	TFAILP			SEPL	OPDECS
		LFMOS	UCL			SEUR	SADECS
		LFMWUC	UF			USEHRS	SBARC
		LSARC	WGTL			XSE	SBDARC
		NABFM					SBRCT
		NPPAFM					SBSARC
		NSERFM					SCRPSC
		NTBLFM					SDARC
		NTDLFM					SDRCTC
		OPDECL					SDRCTO
		SADECL					SEPTS
		SCRPLC					SOB
		SEPTFM					SOD
		SRUPTR					SOS
		TDPLFM					SRUNAM
		TRCLFM					SSARC
		TRWLFM					SSEHR
		UCPPFM					SSEUHB
		WTPPFM					SSEUHD
							SWUC
							TDPS
							TRCS
							TRWS
							UCPPS
							UCS
							WGTS
							WTPPS

IMPLICATIONS FOR DATA FILES. Changes to array sizes may require changes to some file definition control cards and/or file definition statements in the program. Changing the dimensions for LRU failure mode arrays requires a change for file code 15, and changes to SRU array dimensions requires a change for file code 16. Similarly, changes to network node arrays affect the file definition for file code 13, and changes to network arc arrays affect file code 14. Figure B.10 shows the IBM compatible file definition statements, in the main routine of the program, appropriate for default sized arrays and those definition statements modified to be consistent with Figures B.3, B.4, B.7, and B.8.

```
02290 C      SET RECORD SIZE FOR RANDOM FILES 15 & 16
02320 C IBM   FIRST PARAMETER AFTER '(' IS THE DIMENSIONED SIZE FOR --
02330 C IBM     LRU FAILURE MODES, FC-15
02340 C IBM     SRU'S               , FC-16
02350 C     DEFINE FILE 15(40,24,U,N15)
02360 C     DEFINE FILE 16(40,26,U,N16)
02361 C IBM FORTRAN VS STATEMENTS:
02362 C VS   RECL IS 24 * 4 FOR FC-15
02363 C VS   RECL IS 26 * 4 FOR FC-16
02364       OPEN (UNIT=15,STATUS='UNKNOWN',ACCESS='DIRECT',RECL=96,
02365      &   FORM='UNFORMATTED')
02366       OPEN (UNIT=16,STATUS='UNKNOWN',ACCESS='DIRECT',RECL=104,
02367      &   FORM='UNFORMATTED')

10590 C IBM FILE DEFINITIONS FOR RANDOM FILES 13 & 14 -- SECOND PARAM.
10600 C IBM     WITHIN '(' IS MAXNOD+3 & MAXARC, AFTER FIRST RUN
10610 C IBM     CHANGE TO THE LOWER NUMBERS NNODES+3 & NARCS
10620 C
10630 C     DEFINE FILE 13(50,203,U,N13)
10640 C     DEFINE FILE 14(50,400,U,N14)
10650 C IBM FORTRAN VS STATEMENTS:
10651 C VS  RECL FOR FC-13 IS (MAXNOD + 3) * 4
10652 C VS  RECL FOR FC-14 IS MAXARC * 4
10653 C VS  AFTER FIRST RUN, USE NNODES+3 AND NARCS IN FORMULAS
10654       OPEN (UNIT=13,STATUS='SCRATCH',ACCESS='DIRECT',RECL=812,
10655      &   FORM='UNFORMATTED')
10656       OPEN (UNIT=14,STATUS='SCRATCH',ACCESS='DIRECT',
10657      &   RECL=1600,FORM='UNFORMATTED')
```

Figure B.10a. File definition statements for default sizes.

```
02290 C      SET RECORD SIZE FOR RANDOM FILES 15 & 16
02320 C IBM   FIRST PARAMETER AFTER '(' IS THE DIMENSIONED SIZE FOR --
02330 C IBM     LRU FAILURE MODES, FC-15
02340 C IBM     SRU'S               , FC-16
02350 C     DEFINE FILE 15(90,24,U,N15)
02360 C     DEFINE FILE 16(80,26,U,N16)
02361 C IBM FORTRAN VS STATEMENTS:
02362 C VS   RECL IS 24 * 4 FOR FC-15
02363 C VS   RECL IS 26 * 4 FOR FC-16
02364       OPEN (UNIT=15,STATUS='UNKNOWN',ACCESS='DIRECT',RECL=96,
02365      &   FORM='UNFORMATTED')
02366       OPEN (UNIT=16,STATUS='UNKNOWN',ACCESS='DIRECT',RECL=104,
02367      &   FORM='UNFORMATTED')

10590 C IBM FILE DEFINITIONS FOR RANDOM FILES 13 & 14 -- SECOND PARAM.
10600 C IBM     WITHIN '(' IS MAXNOD+3 & MAXARC, AFTER FIRST RUN
10610 C IBM     CHANGE TO THE LOWER NUMBERS NNODES+3 & NARCS
10620 C
10630 C     DEFINE FILE 13(50,358,U,N13)
10640 C     DEFINE FILE 14(50,1270,U,N14)
10650 C IBM FORTRAN VS STATEMENTS:
10651 C VS  RECL FOR FC-13 IS (MAXNOD + 3) * 4
10652 C VS  RECL FOR FC-14 IS MAXARC * 4

10653 C VS  AFTER FIRST RUN, USE NNODES+3 AND NARCS IN FORMULAS
10654       OPEN (UNIT=13,STATUS='SCRATCH',ACCESS='DIRECT',RECL=1432,
10655      &   FORM='UNFORMATTED')
10656       OPEN (UNIT=14,STATUS='SCRATCH',ACCESS='DIRECT',
10657      &   RECL=5080,FORM='UNFORMATTED')
```

Reproduced from best available copy.

Figure B.10b. File definition statements for increased sizes.

Figure B.10. File definition statements.

APPENDIX C. Additional Dimensioning Requirements

Introduction. This appendix was prepared to supplement Appendix B, NRLA Program Array Dimensions.

Programs Updated with Changes #5. If your copy of the NRLA program has been updated with NRLA Changes #5, February 1984, an additional dimensioning requirement exists.

The array SERN must be dimensioned to MAXSE + 16. In the default case, the SE arrays are dimensioned to MAXSE = 20. Thus, the default size for SERN is 36.

If the SE arrays were increased to 35 as in fig B.1, page 62, the dimensioned size for SERN must be increased to 51 as shown below.

SERN with default size:

```
1090      INTEGER C1,C2,C12,C2PREV,SEN,SERN(36)
```

SERN with increased size to match MAXSE = 35:

```
1090      INTEGER C1,C2,C12,C2PREV,SEN,SERN(51)
```

As with the other dimensioning changes, this change must be made 14 times; once in the main routine, once in Block Data, and once in every subroutine except SORT.

Programs Updated with Changes #6. If your copy of the NRLA program has been updated with NRLA Changes #6, May 1984, the dimensioning process has been greatly simplified. To redimension, only the PARAMETER statements need to be modified. These PARAMETER statements are located in the main routine, in BLOCK DATA, and in every subroutine except SORT.

The following example shows redimensioning the program from the default dimensions to accomodate 35 SE, 40 LRUs, 90 LRU failure modes, 80 SRUs, a cross-reference size of 500, 355 nodes, and 1270 arcs. (See Appendix B for how to determine array sizes required.)

PARAMETER statements with default values:

```
1035C      FOLLOWING ARE THE PARAMETER STATEMENTS
1036C     TO ADJUST DIMENSIONING REQUIREMENTS SIMPLY CHANGE
1037C     THESE VALUES IN ALL SUBROUTINES
1038C
1039      PARAMETER (MAXLRU=25,MAXFM=40,MAXSRU=40)
1040      PARAMETER (MAXSE=20,MAXREF=210,MAXITM=210)
1041      PARAMETER (MAXNOD=200,MAXARC=400,MARCP2=402)
1042      PARAMETER (MAXSEP=36)
1043C
```

PARAMETER statements with increased array sizes:

```
1035C      FOLLOWING ARE THE PARAMETER STATEMENTS
```

```
1036C     TO ADJUST DIMENSIONING REQUIREMENTS SIMPLY CHANGE
1037C     THESE VALUES IN ALL SUBROUTINES
1038C
1039      PARAMETER (MAXLRU=80,MAXFM=90,MAXSRU=40)
1040      PARAMETER (MAXSE=35,MAXREF=500,MAXITM=500)
1041      PARAMETER (MAXNOD=355,MAXARC=1270,MARCP2=1272)
1042      PARAMETER (MAXSEP=51)
1043C
```

Dimensioning NRLA Without Use of PARAMETER Statements. If your FORTRAN does not support PARAMETER statements, the following procedure should be followed to dimension NRLA.

1. Following the guidelines in Appendix B of the NRLA Programmer's Guide, determine the array sizes needed to fulfill your program requirements.

2. Delete or comment out all the PARAMETER statements.

3. Make the appropriate changes to the program arrays. Using global changes, change all occurrences of the specified string to the numbers computed in Step 1 as follows:

Change all occurrences of	to the maximum number of
(MAXLRU	LRUs
(MAXFM	Failure modes
(MAXSRU	SRUs
(MAXSE	Support Equipment (SE)
(MAXSEP	MAXSE + 16
(MAXITM	SE specified on "32, 42, or 52" cards
(MAXREF	SE to item relationships
(MAXNOD	Network nodes
(MAXARC	Network arcs
(MARCP2	MAXARC + 2

NOTE: Changing all occurrences of the string without the "(" in front of it, will result in some changes being made that should have remained unchanged. For example, if all occurrences of "MAXLRU" instead of "(MAXLRU" were changed to "40", the line "IF (NLRU.LE.MAXLRU) GO TO 320" would have been changed to "IF (NLRU.LE.40) GO TO 320". As long as you never redimensioned the model, this extra change would be fine. However, if you do need to redimension the model, this line could be easily overlooked and could cause errors.

4. To make statements such as "IF (NLRU.LE.MAXLRU) GO TO 320" executable, DATA statements must be inserted in the program to give these variables values. For example, if you had dimensioned NRLA to have 40 LRUs, 90 failure modes, 80 SRUs, 35 SE, 500 SE specified on "32, 42, or 52" cards, 500 SE to item relationships, 355 nodes, and 1270 arcs, the following statements must be inserted in each subroutine:

```
DATA MAXLRU,MAXFM,MAXSRU/40,90,80/
DATA MAXSE,MAXSEP,MAXITM,MAXREF/35,51,500,500/
DATA MAXNOD,MAXARC,MARCP2/355,1270,1272/
```

END

FILMED

10-85

DTIC